

**AUTOMATING LIDAR DATA TO
DEVELOP AND MANAGE ACTIVE
TRANSPORTATION ASSET
INVENTORIES**

Final Report

SPR 850



Oregon Department of Transportation

AUTOMATING LIDAR DATA TO DEVELOP AND MANAGE ACTIVE TRANSPORTATION ASSET INVENTORIES

Draft Final Report

SPR-850

Submitted by

Michael J. Olsen, PhD, Professor

Jaehoon Jung, PhD, Assistant Professor (Senior Research)

Erzhuo Che, PhD, Assistant Professor (Senior Research)

Heidar Rastiveis, Assistant Professor (Senior Research)

Yelda Turkan, PhD, Associate Professor

Christopher Parrish, PhD, Associate Professor

Oregon State University

101 Kearney Hall, Corvallis OR 97330

for

Oregon Department of Transportation

Research Section

555 13th Street NE, Suite 1

Salem OR 97301

and

Federal Highway Administration

1200 New Jersey Avenue SE

Washington, DC 20590

July 2026

1. Report No. FHWA-OR-RD-27-03	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle Automated Lidar Data to Develop and Manage Active Transportation Asset Inventories		5. Report Date July 2026	
		6. Performing Organization Code	
7. Author(s) Jaehoon Jung, Michael J. Olsen, Erzhua Che, Heidar Rastiveis, Yelda Turkan, and Chris Parrish		8. Performing Organization Report No.	
9. Performing Organization Name and Address Oregon State University 101 Kearney Hall Corvallis, OR 97331		10. Work Unit No. (TRAIS)	
		11. Contract or Grant No.	
12. Sponsoring Agency Name and Address Oregon Dept. of Transportation Research Section 555 13 th Street NE, Suite 1 Salem, OR 97301 Federal Highway Admin. 1200 New Jersey Avenue SE Washington, DC 20590		13. Type of Report and Period Covered Final Report	
		14. Sponsoring Agency Code	
15. Supplementary Notes			
16. Abstract Pedestrian crosswalks, bike lanes, and medians are key safety devices in a multi-modal highway network. However, there is no statewide map of these assets' locations and types. Such a map would benefit planning, maintenance, safety assessments, and network analyses for ODOT and its partners. Currently, ODOT collects asset information through site visits and surveys, which are labor-intensive and costly across the state's vast road network. GIS databases are created ad-hoc and inconsistently, often by interns. Fortunately, ODOT Engineering Automation regularly collects 3D mobile lidar data, providing geometric and radiometric information. This research reviewed state-of-the-art approaches for detecting and classifying pedestrian crosswalks, bike lanes, and medians, and developed a methodology to extract these features from LiDAR datasets for updating ODOT's GIS platform. The approach was tested on several datasets and showed good extraction performance, requiring minimal manual cleanup. Lastly, a plan is provided for ODOT to implement this research.			
17. Key Words Mobile lidar, Photogrammetric data, Crosswalk, Bike Lane, Median, Classification		18. Distribution Statement Copies available from NTIS, and online at 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 214	22. Price

SI* (Modern Metric) Conversion Factors
Approximate Conversions to SI Units

Physical Quantity	Symbol	When You Know	Multiply By	To Find	Symbol
Length	n	inches	25.4	millimeters	mm
Length	ft	feet	0.305	meters	m
Length	yd	yards	0.914	meters	m
Length	mi	miles	1.61	kilometers	km
Area	in ²	square inches	645.2	square millimeters	mm ²
Area	ft ²	square feet	0.093	square meters	m ²
Area	yd ²	square yard	0.836	square meters	m ²
Area	ac	acres	0.405	hectares	ha
Area	mi ²	square miles	2.59	square kilometers	km ²
Volume	fl oz	fluid ounces	29.57	milliliters	mL
Volume	gal	gallons	3.785	liters **	L
Volume	ft ³	cubic feet	0.028	cubic meters	m ³
Volume	yd ³	cubic yards	0.765	cubic meters	m ³
Mass	oz	ounces	28.35	grams	g
Mass	lb	pounds	0.454	kilograms	kg
Mass	T	short tons (2000 lb)	0.907	megagrams (or "metric ton")	Mg (or "t")
Temperature (exact degrees)	oF	Fahrenheit	5 (F-32)/9 or (F-32)/1.8	Celsius	oC
Illumination	fc	foot-candles	10.76	lux	lx
Illumination	fl	foot-Lamberts	3.426	candela/m ²	cd/m ²
Force and Pressure or Stress	lbf	poundforce	4.45	newtons	N
Force and Pressure or Stress	lbf/in ²	poundforce per square inch	6.89	kilopascals	kPa

*SI is the symbol for the International System of Measurement

** Volumes greater than 1000 L shall be shown in m³

SI* (Modern Metric) Conversion Factors
Approximate Conversions from SI Units

Physical Quantity	Symbol	When You Know	Multiply By	To Find	Symbol
Length	mm	millimeters	0.039	inches	in
Length	m	meters	3.28	feet	ft
Length	m	meters	1.09	yards	yd
Length	km	kilometers	0.621	miles	mi
Area	mm ²	square millimeters	0.0016	square inches	in ²
Area	m ²	square meters	10.764	square feet	ft ²
Area	m ²	square meters	1.195	square yards	yd ²
Area	ha	hectares	2.47	acres	ac
Area	km ²	square kilometers	0.386	square miles	mi ²
Volume	mL	milliliters	0.034	fluid ounces	fl oz
Volume	L	liters	0.264	gallons	gal
Volume	m ³	cubic meters	35.314	cubic feet	ft ³
Volume	m ³	cubic meters	1.307	cubic yards	yd ³
Mass	g	grams	0.035	ounces	oz
Mass	kg	kilograms	2.202	pounds	lb
Mass	Mg (or "t")	megagrams (or "metric ton")	1.103	short tons (2000 lb)	T
Temperature (exact degrees)	oC	Celsius	1.8C+32	Fahrenheit	oF
Illumination	lx	lux	0.0929	foot-candles	fc
Illumination	cd/m ²	candela/m ²	0.2919	foot-Lamberts	fl
Force and Pressure or Stress	N	newtons	0.225	poundforce	lbf
Force and Pressure or Stress	kPa	kilopascals	0.145	poundforce per square inch	lbf/in ²

For More Information see: <https://www.fhwa.dot.gov/publications/convtabl.cfm>

ACKNOWLEDGEMENTS

The research team appreciates the efforts and insights of the research coordinator, Josh Roll, and the TAC in providing feedback on the project. Rhonda Dodge and Jonathan Rawlings were very helpful in providing mobile lidar data for the areas of interest and for testing the tools and outputs. Jennifer Lanzarotta assisted with the tool testing. Dr. Suhong from GNU assisted with the road marking clustering algorithm. Lael Case and Tyler Russel, undergraduate research assistants at OSU performed the data labeling and annotation.

DISCLAIMER

This document is disseminated under the sponsorship of the Oregon Department of Transportation and the United States Department of Transportation in the interest of information exchange. The State of Oregon and the United States 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 or the United States Department of Transportation.

The State of Oregon and the United States 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.

Drs. Olsen and Che have financial interests in EZDataMD LLC, a company which commercializes point cloud analysis technology related to this research. The conduct, outcomes, or reporting of this research could benefit EZDataMD LLC and could potentially benefit Drs. Olsen and Che.

TABLE OF CONTENTS

1.0	INTRODUCTION	1
1.1	PROBLEM STATEMENT	1
1.2	OBJECTIVES OF THE STUDY	2
1.3	ORGANIZATION OF THE REPORT	3
2.0	LITERATURE REVIEW	5
2.1	FEATURE TYPES	5
2.1.1	<i>Bike lanes</i>	5
2.1.2	<i>Crosswalks</i>	6
2.1.3	<i>Medians</i>	7
2.2	OREGON DOT PROCEDURES	8
2.2.1	<i>Inventory protocols</i>	8
2.2.2	<i>Linear Referencing Systems</i>	9
2.2.3	<i>TransGIS</i>	11
2.3	MOBILE LIDAR TECHNOLOGY	14
2.3.1	<i>Mobile Lidar System Applications</i>	14
2.3.2	<i>Mobile Lidar System Basic Technical Information</i>	15
2.3.3	<i>Mobile Lidar at Oregon DOT</i>	18
2.3.4	<i>Mobile Lidar Standards, Specifications and Guidelines</i>	19
2.4	PHOTOGRAMMETRIC DATA	20
2.4.1	<i>Photogrammetry Basics</i>	20
2.4.2	<i>Road Extraction via Photogrammetry</i>	21
2.5	OBJECT EXTRACTION METHODS	22
2.5.1	<i>Pre-processing</i>	22
2.5.2	<i>Segmentation</i>	30
2.5.3	<i>Classification</i>	38
2.6	ATTRIBUTION AND ANALYSIS	46
2.7	SUMMARY	46
3.0	METHODOLOGY OVERVIEW	49
3.1	OVERVIEW	49
3.2	TOOL DEVELOPMENT	51
4.0	TRAINING DATASET	55
4.1	SITE SELECTION	55
4.2	DATA ANNOTATION	63
4.2.1	<i>Categories</i>	63
4.2.2	<i>Attribution</i>	63
4.2.3	<i>Data Files</i>	63
4.2.4	<i>Procedure</i>	69
4.2.5	<i>Utilization in machine learning</i>	69
5.0	DATA PREPARATION	73
5.1	DATA ORGANIZATION AND INDEXING	73
5.2	DATA PREPROCESSING	75
5.2.1	<i>Profile density estimation</i>	75
5.2.2	<i>Ground filtering</i>	76

5.2.3	<i>Curb extraction</i>	77
5.2.4	<i>Road Surface Extraction</i>	80
6.0	FEATURE EXTRACTION ALGORITHMS	81
6.1	ROAD MARKING EXTRACTION	81
6.2	ROAD MARKING CLUSTERING	82
6.3	ROAD LANE EXTRACTION	86
6.4	RAISED MEDIAN AND STAGGERED CONTINENTAL EXTRACTION	88
6.5	STANDARD PARALLEL EXTRACTION	90
6.6	BIKE LANE EXTRACTION	92
7.0	TESTING RESULTS	93
7.1	TRANSGIS COMPARISON	93
7.2	QUALITATIVE EVALUATION	94
7.3	TESTING RESULTS	95
7.3.1	<i>Bike lanes</i>	95
7.3.2	<i>Crosswalks</i>	103
7.3.3	<i>Medians</i>	111
7.3.4	<i>Road marking width evaluation</i>	117
7.4	PROCESSING TIME EVALUATION	117
7.4.1	<i>Time savings</i>	119
7.4.2	<i>Lane marking time savings</i>	120
8.0	IMPLEMENTATION	123
8.1	TOOL TESTING	124
8.2	SOFTWARE USAGE	125
9.0	CONCLUSIONS	127
10.0	REFERENCES	129
APPENDIX A: MILEPOST INDEXING TOOL		A-1
A.1	INTRODUCTION	A-1
A.2	RUNNING THE SCRIPT	A-1
A.2.1	<i>Additional Notes</i>	A-2
A.3	QUERIES TO IDENTIFY DATASETS	A-2
APPENDIX B: USER MANUAL CURB EXTRACTION		B-1
B.1	OVERVIEW	B-2
APPENDIX C: USER MANUAL: ROME		C-1
C.1	DISCLAIMER	C-1
C.2	INTRODUCTION	C-2
C.3	INTENDED AUDIENCE	C-2
C.4	PARTICIPANTS	C-3
C.5	PUBLICATIONS	C-3
C.6	GETTING STARTED	C-4
C.7	LICENSING	C-4
C.8	OVERVIEW	C-4
C.9	TOOL FEATURES	C-5

C.10 INPUT FILES	C-9
C.11 OUTPUT FILES.....	C-9
<i>C.11.1 GIS Shapefiles.....</i>	<i>C-9</i>
<i>C.11.2 CSV Files (Legacy)</i>	<i>C-12</i>
C.12 GIS DATABASE.....	C-15
C.13 RECOMMENDATIONS	C-23
C.14 REFERENCES.....	C-24
C.15 APPENDIX.....	C-26
APPENDIX D: TOOL EVALUATION RUBRIC	D-1
APPENDIX E: FEATURE EXTRACTION SUMMARY	E-4

LIST OF TABLES

Table 2-1: Summary of studies in ground extraction from point clouds	23
Table 2-2: Summary of road surface extraction methods	26
Table 2-3: Summary of road marking segmentation methods	34
Table 2-4: Summary of road marking classification methods.....	42
Table 3-1. Tools provided with SPR850 and associated inputs and outputs.....	52
Table 4-1: Proposed sites to support the algorithm training and testing.....	56
Table 4-2: Feature types and relevant attributes	65
Table 4-3: Occurrences of features in the explored datasets showing the counts of available features in all files as well as only the labeled files.	66
Table 4-4: Occurrences of features in the labeled datasets	67
Table 7-1. Summary of the primary datasets used for the testing and evaluation.....	93
Table 7-2. Summary performance metrics for bike lane extraction and observations from the qualitative analysis.	97
Table 7-3. Summary performance metrics for crosswalk extraction and observations from the qualitative analysis.	104
Table 7-4. Summary performance metrics for median extraction and observations from the qualitative analysis.	112
Table 7-5. Summary of marking width accuracy assessment.	117
Table 7-6. Summary of processing time required for executing the SPR850 tools.....	118
Table 7-7. Comparison of processing time required for manual extraction in Civil3D.	119
Table 7-8. Time savings computations based on the processing times in Table 7-6 and Table 7-7.	120
Table 7-9. Lane marking time savings from Rome. Values from Markovic et al. (2024) were scaled to a 1 km section.	121
Table 8-1. Tasks associated with training for research products.....	125
Table B-1. Executables called for the curb extraction process*.....	B-2
Table B-2. ArcPy Script functions.....	B-3
Table C-1. Description of the features in the main GUI shown in Figure C-1 and C-2	C-7
Table C-2. Example format for the trajectory data file (asciitry).....	C-9
Table C-3. Example of fields and data provided in the feature shapefile	C-11
Table C-4. Example of fields and data in the Run table	C-13
Table C-5. Example of fields and data in the Section table	C-13
Table C-6. Example of fields and data provided in the Stripe table	C-14
Table C-7. Example of fields and data provided in the Node table.....	C-14
Table C-8. Example of the fields and data provided in the Retro table.....	C-15
Table C-9. Description of data layers produced with the python script tool.....	C-16
Table C-10. Grade (Condition Score) thresholds	C-21
Table D-1. Summary of datasets explored for the evaluation.	D-1
Table D-2. Rubric for the evaluation.	D-1
Table E-1. Summary of features in all data files considered for training data.	E-4

LIST OF FIGURES

Figure 2-1: Types of bikeway: (a) bike lane with stencils and stripes; (b) shoulder bikeway; and (c) shared lane (ODOT, 2020)	5
Figure 2-2: The shared lane marking (sharrow) symbol (ODOT, 2020)	6
Figure 2-3: Types of buffered bike lanes (FHWA 2015).....	6
Figure 2-4: Types of crosswalk: (a) standard parallel; (b) staggered continental; and (c) ladder (ODOT, 2020).....	7
Figure 2-5: Patterns of crosswalk: (a) standard crosswalk and (b) staggered continental crosswalk (ODOT, 2020).....	7
Figure 2-6: Locations of crosswalk: (a) intersection and (b) mid-block (ArcGIS Pro imagery)	7
Figure 2-7: Example of gravel median in 3D point cloud data and its profile view	8
Figure 2-8: Overview of example datasets in TransGIS relevant to this study.....	12
Figure 2-9: Example staggered continental crosswalk in TransGIS.....	12
Figure 2-10: Example standard parallel crosswalk in TransGIS.....	13
Figure 2-11: Example bike lane in TransGIS.....	13
Figure 2-12: Example traffic barriers layer in TransGIS	14
Figure 2-13: Sample applications using mobile lidar technology in transportation. From NCHRP Report 748 (Olsen et al. 2013).....	15
Figure 2-14: Mobile lidar data collected by ODOT on Highway OR-231 at the Spangler Landslide, near Molalla, OR. In the top image, the lidar data are colored using the imagery collected from the camera integrated in the mobile lidar system. In the middle image, the same point cloud is shown displayed by intensity. Although the intensities shown here are un-calibrated, pavement markings and other features are clearly distinguishable. The bottom image shows the point cloud color-coded by elevation, where colors are alternated every 0.20 m increase in elevation.....	17
Figure 2-15: Oregon DOT’s current mobile lidar system, Leica Pegasus:Two	18
Figure 2-16: Stereo photogrammetry: aerial imagery acquisition schematic, showing a pair of overlapping images of the OSU campus.	21
Figure 2-17: Example of <i>Vo-SmoG</i> ground filtering on a mobile lidar data collected by ODOT with the Leica Pegasus:Two system: (a) Original and (b) Ground points.....	25
Figure 2-18: Example of the height jump along the curb, serving as a road boundary in point cloud data.	29
Figure 2-19: Example of <i>Vo-Norvana</i> segmentation on MLS dataset collected by ODOT using Leica Pegasus:Two where each color represents a segment while the unclassified points are colored in black.....	32
Figure 2-20: Example results from the ROME tool (ver. 2.1): (a) segmented road markings and (b) RoME Toolbox and its associated scripts within ArcCatalog.....	37
Figure 2-21: Classification of common urban objects within a point cloud collected by ODOT’s Leica Pegasus:Two.....	45
Figure 3-1. Overview of the tools, workflows, and methodologies utilized in SPR850.....	50
Figure 3-2: ROME v3.7 GUI developed by the research team. See Appendix C for more details.	53
Figure 4-1: Data Collection Site #1 in Downtown Salem, Oregon consisting of an urban area between mileposts 16 – 21 on Highway 221 as well as a section of Highway 99 EB between mileposts 0.0 to 8.5.....	57

Figure 4-2: Data Collection Site #1 in Tigard Oregon consisting of mileposts 8 to 13 on Highway 99W, and from mileposts 2 to 7 on Highway 141. This site includes communities in Tigard and King City.....	58
Figure 4-3: Data collection site #3 between mileposts 0 and 18 on Tualatin Highway No. 029, and includes a section of Nehalem Highway No. 102, between milepost 88 to the intersection with Tualatin Highway No. 029. This site includes communities in Forest Grove, Hillsboro, and Beaverton.....	59
Figure 4-4: Data collection site #4. Site#4 consisting of mileposts 2 to 8 on Highway 99E, and the sections of Santiam Highway 16 and Albany – Junction City Highway between I5 and 99E.....	60
Figure 4-5: Data collection sites #5 on Highway 213 (82 nd Ave) in Portland, OR.	61
Figure 4-6: The alternate site is located on Oregon Coast Highway 101 between mileposts 134 and 140.....	62
Figure 4-7: Summary of pedestrian features common on ODOT highways. Lower priority features are either less common or more difficult to extract.	64
Figure 4-8: Example annotated point cloud with each class of marking identified with a unique color.....	70
Figure 4-9: Example annotated point cloud with each class of marking identified with a unique color. (Annotated features only).	71
Figure 4-10: Example errors in initial plan view annotation that were corrected.	71
Figure 5-1: Workflow for lidar milepost indexing.	74
Figure 5-2: Visualization of the misalignment between the MLS trajectory and polylines generated from the lower accuracy milepost geodatabase.....	74
Figure 5-3: Alignment between the MLS trajectory and the milepost geodatabase.....	75
Figure 5-4: Example of profile density estimation for mobile lidar data.....	76
Figure 5-5: Example curb detection results: (a) MLS data in RGB color; (b) Ground filtering result from <i>Vo-SmoG</i> where green points represent ground points; and (c) curb detection result (red points)	78
Figure 5-6. Example curb extraction. Left: Full dataset, Right Extrated curb points only.....	79
Figure 6-1. Example road marking extraction (green points) for two different highways.	82
Figure 6-2. Example of the road marking clustering by: (a) connected component labeling method and (b) the proposed approach.....	83
Figure 6-3. Example of the improvements provided by the road marking clustering algorithm. The left shows the point cloud colored by RGB with vehicles blocking the view of the scanner. The middle shows the initial clustering (each color identifies a unique marking) where the marking is separated at the data gaps. The right shows the corrected marking where the clusters belonging to the same marking are linked.	84
Figure 6-4. Examples of road marking clustering. The top row shows examples of successful linkage of markings across data gaps or dotted lines. The bottom row shows areas where the clustering algorithm was not successful due to connections between markings, (1), large data gaps (13 and 4), or high curvature or wear (6).	85
Figure 6-5. Conceptual image for identifying individual lanes using the extracted road markings.	87
Figure 6-6. Example of road lane extraction (white lines): The height of the road lanes has been adjusted to match the vehicle trajectory height for visualization purposes.	87

Figure 6-7. Example of raised median and marked crosswalk extraction: (a) the point cloud in RGB colors and (b) the extracted features in discrete colors.....	89
Figure 6-8. Illustration of standard parallel markings: (a) the point cloud data in RGB colors and (b) the segmented markings highlighted in yellow.	91
Figure 6-9. Example of bike lane extraction (white line): The height of the bike lanes has been adjusted to match the vehicle trajectory height for visualization purposes.	92
Figure 7-1. Bike lane extraction results for the evaluation section selected from Dataset 13 showing true positives, false negatives, and false positives. In this section, the false positives tend to correspond with intersections and are connections between the bike lane segments. On the left side a small portion of the bike lane is missed; however, it is on the opposite side of the road from the mobile lidar trajectory where the data were sparse due to occlusions.....	98
Figure 7-2. Bike lane extraction results for the evaluation section selected from Dataset 23 showing true positives, false negatives, and false positives. In this section, the entire bike lane was missed. It was, however, extracted as a vehicle travel lane.	99
Figure 7-3. Bike lane extraction results for the evaluation section selected from Dataset 15 showing true positives, false negatives, and false positives. RoME has successfully extracted a bike lane in the occluded area (no data) using its post-processing mechanism.	100
Figure 7-4. Bike lane extraction results for the evaluation section selected from Dataset 32 showing true positives, false negatives, and false positives. The false negatives on the left are a result of occlusions from vehicles. The false negatives on the right are likely due to multiple curbs extracted.....	101
Figure 7-5. Bike lane extraction results for the evaluation section selected from Dataset 2 showing true positives, false negatives, and false positives. The bike lane on the crossroad was missed because ROME only extracts bike lanes parallel to the trajectory.	102
Figure 7-6. Crosswalk extraction results for the evaluation section selected from Dataset 13 showing true positives, false negatives, and false positives of the bike lane. While portions of the crosswalks are missing, ROME still identifies the presence of a crosswalk.	105
Figure 7-7. Crosswalk extraction results for the evaluation section selected from Dataset 23 showing true positives, false negatives, and false positives. Some false positives occur in a noisy section of the data on rumble strips. Most crosswalks are extracted; however a portion of the cross walk near the median was missed.....	106
Figure 7-8. Crosswalk extraction results for the evaluation section selected from Dataset 15 showing true positives, false negatives, and false positives. Most portions of the crosswalk were correctly identified. Most false negatives were extracted as lane markings but not properly classified as crosswalks.	107
Figure 7-9. Crosswalk extraction results for the evaluation section selected from Dataset 11 showing true positives, false negatives, and false positives. ROME struggled to extract the smaller crosswalk sections near the medians.	108
Figure 7-10. Crosswalk extraction results for the evaluation section selected from Dataset 3 showing true positives, false negatives, and false positives.	109
Figure 7-11. Crosswalk extraction results for the evaluation section selected from Dataset 2 showing true positives, false negatives, and false positives.	110
Figure 7-12. Median extraction results for the evaluation sections selected from Datasets 23, 3, and 15 showing True positive, false negatives, and false positives.	113

Figure 7-13. Median extraction results for the evaluation section selected from Dataset 32 showing true positives, false negatives, and false positives.	114
Figure 7-14. Median extraction results for the evaluation section selected from Dataset 2 showing true positives, false negatives, and false positives. In this case, a portion of the median was still captured.....	115
Figure 7-15. Example output of the curb extraction where medians can be clearly identified from the curb extraction.....	116
Figure A-0-1. Example of running the script from the power shell.....	A-2
Figure A-0-2. Example of data query using the milepost indexing table (HWYNAME: PACIFIC, MP: 252 - 254).	A-4
Figure B-0-1. Running the second script to generate curb polylines from the rasters.....	B-5
Figure B-0-2. Running the second script in batch to generate curb polylines from the rasters. .	B-6
Figure B-0-3. Example output feature class of curblines in ArcGIS Pro.....	B-6
Figure C-0-1. Main GUI of the ROME program (version 3.7)	C-5
Figure C-0-2. Interface for selecting features for GIS layer outputs	C-6
Figure C-0-3. Example screenshot of the output layers in a geodatabase in ArcGIS Pro.	C-11
Figure C-0-4. Screenshot of the output layers in a geodatabase in ArcCatalog	C-17
Figure C-0-5. GUI for the csv2gdb.py tool in ArcGIS Pro showing input parameters as well as command line output.	C-18
Figure C-0-6. Example output of the csv2gdb.py script showing attributes available in the stripe table.	C-18
Figure C-0-7. GUI for the stripeEditor.py script.	C-19
Figure C-0-8. Example output of the stripeEditor.py script showing correct yellow and white stripes with transverse stripes removed.	C-20
Figure C-0-9. GUI for the retrograder.py script and associated inputs in ArcGIS Pro	C-22
Figure C-0-10. Example output of the retrograder script showing the MP_retro results for a single northbound and single southbound pass colored by overall condition score.	C-23
Figure C-0-11. Internal parameters for ROME v3.7.....	C-26

1.0 INTRODUCTION

1.1 PROBLEM STATEMENT

Pedestrian crosswalks, bike lanes, and medians are important traffic devices that facilitate safe navigation within a multi-modal transportation network for drivers, pedestrians, and bicyclists. Currently, there is no comprehensive statewide map of the locations and types (e.g., midblock crossing) of these assets; however, such a map\GIS layer would be very useful for planning, maintenance, systematic safety studies, and many types of network analyses that are important to ODOT and its partner agencies at the local level. Site visits and surveys to capture this information are time-consuming and costly for the vast road network across the state that ODOT and its partners maintain. Fortunately, ODOT Geomtronics\Engineering Automation collects 3D mobile lidar data on a two-year cycle, providing rich geometric and radiometric information of the ODOT's road network. Unfortunately, despite this rich source of data, current workflows to process these mobile lidar data require manual extraction of these assets as Microstation or GIS objects using specialized software. More efficient approaches to extract this information are necessary in order to create a statewide inventory of certain roadway assets needed by transportation engineers. Such approaches enable ODOT to obtain additional returns on investment (ROI) with the mobile lidar system currently used for several applications with minimal cost.

Unlike the other types of road features, there have been few research studies completed to date on the detection and classification of pedestrian crosswalks, bike lanes, and medians due to many difficulties in such tasks associated with their complex shapes, occlusions, and noise in sensor data. Several image-based methods for the detection of bike lanes and pedestrian crosswalks have been proposed; however, accurate geo-positioning and sensitivity to light conditions are still limitations to overcome (Lausser et al. 2008; Boudet and Midenet, 2009; Hoang et al. 2019). Automated detection of pedestrian crosswalks using mobile lidar data was first introduced by Riveiro et al. (2015), where the point cloud is rasterized onto a 2D binary image to apply an image processing technique (Hough Transform) to identify the locations of a series of parallel lines as a pedestrian crosswalk. Unfortunately, the approach does not provide segmented crosswalks.

With recent breakthroughs in deep learning, some approaches that can learn semantic, high-level deep features, have demonstrated high performance in classifying various types of road markings (Ma et al. 2020). For example, Liu et al. (2020) proposed a pix2pix deep learning framework for semantic segmentation of 10 types of road marking including pedestrian crosswalks. However, deep-learning-based approaches have limitations when sufficient training data are not available or when attempting to extend them to work in a different scene from those captured in the training dataset. Further, designing the network architecture still depends heavily on a heuristic approach requiring users to experiment to determine appropriate parameters for their specific data, limiting its scalability. Accordingly, simplified rule-based approaches that apply geometric or radiometric thresholds are still widely adopted. For example, Soilán et al. (2017) first classify

rectangular-shaped markings from others and then identify a group of parallel rectangular markings by evaluating the azimuth angle with respect to the trajectory and the distance between the centroids of each rectangular markings. Wen et al. (2019) propose a multi-scale clustering method to classify lanes and crosswalks based on the fact that they typically have a longer length and repetitive pattern compared with other types of markings. Kuma et al. (2018) use airborne lidar data to extract the initial median points by applying the elevation and intensity threshold parameters, which are then rasterized into a binary image to apply a morphological closing operation for noise removal.

While these techniques are promising, they generally have not yet advanced beyond the research stage and are not ready for large scale implementation. Several machine learning-based feature extraction commercial software products have been made available in recent years. However, these tools still require significant effort to produce training datasets for many types of transportation assets and mobile lidar systems. It is also not clear how much of the workflow is truly automated as most of this work is done as a service where automated approaches are followed by manual cleanup. Alternatively, the applied research presented in SPR850 consists of a comprehensive approach that combines simple rule-based methods with deep learning methods to improve the extraction of the pedestrian crosswalks, bike lanes, and medians using mobile lidar point cloud data possibly in conjunction with imagery. SPR850 leverages several algorithms that use mobile lidar data to automatically extract various road features, such as road markings, surface, and curbs developed through previous research projects completed by the OSU research team for ODOT, PacTrans, and the National Science Foundation (NSF) projects. The use of those algorithms provided a framework to reduce trial and error processes while enabling the development of an innovative approach to the extraction of a broader range of complex road features that have been scarcely studied. The developed approach was developed and tested on a variety of mobile lidar datasets collected by ODOT, which will improve the robustness of the developed approach.

1.2 OBJECTIVES OF THE STUDY

This research explores the capabilities of mobile lidar data and develops automated algorithms and efficient workflows to extract pedestrian crosswalks, bike lanes, and medians with important attributes such that they can be used in asset management, transportation safety, and maintenance applications. Specifically, the primary objectives of this research are to:

- Investigate state-of-the-art machine learning and object detection algorithms for extraction of transportation assets from mobile lidar and photogrammetric data.
- Develop procedures for extracting pedestrian crosswalks, bike lanes, and medians from Oregon DOT's mobile lidar data.
- Produce a GIS geodatabase for several pilot corridors identifying the locations of bike lanes, crosswalks, and medians to support decision making and integrate data analysis results into Oregon DOT's overall workflows. These corridors were selected from urban and rural regions with varying levels of complexity.

- Evaluate benefits (e.g., time or cost savings, inventory completeness, and improved accuracy) associated with the developed approaches compared with current processes.

The products of this research tie directly into ODOT's mission by providing the capability to develop databases with detailed information on important traffic control devices that are critical to the safety of bicyclists and pedestrians in a multi-modal transportation environment and improving safety of ODOT personnel by minimizing the need to acquire this information in the field. This database provides important information to guide decision making, planning, and maintenance efforts to improve the safety and mobility of the transportation network. By leveraging the existing data collected on a regular basis with the mobile lidar system, this research can reduce the significant time and cost required for collecting data about traffic safety devices. Further, the workflows to efficiently extract this information reduce manual processing, resulting in a higher ROI on ODOT's mobile lidar system. Lastly, the framework integrating the tools developed in this research and from the prior work can be extended to enable additional algorithms to be created with significantly reduced effort to extract additional features of interest such as the work underway in SPR866 extracting road surface characteristics such as grade, cross slope, and curvature.

1.3 ORGANIZATION OF THE REPORT

The research report is organized as follows:

- Chapter 2 provides a literature review discussing current ODOT practices, mobile lidar technology, aerial photogrammetry, and object extraction algorithms.
- Chapter 3 presents an overview of the methodology employed in this research and tools developed to implement the methodology and the GIS output layers generated.
- Chapter 4 describes the data annotation procedures to create an extensive training dataset to train the deep learning models utilized in the research as well as provide validation.
- Chapter 5 discusses the workflows implemented for data preparation and pre-processing including mobile lidar data indexing, road extraction, and curb extraction.
- Chapter 6 presents the classification and extraction algorithms developed for identifying crosswalks, bike lanes, and medians.
- Chapter 7 explores the robustness of the algorithms by providing testing results to evaluate the effectiveness and efficiency of the methodology.
- Chapter 8 presents an implementation plan as well as describes potential benefits.
- Chapter 9 outlines the conclusions from this research and opportunities for future research.
- Appendices A-C provide user manuals associated with the tools.

- Appendix D provides the rubric used for evaluation of the tool by ODOT personnel.
- Appendix E contains a summary of features available in the datasets.

2.0 LITERATURE REVIEW

This literature review evaluates the current state-of-the-art methodologies regarding the detection and classification of the pedestrian crosswalks, bike lanes, and medians, with a focus on relevant studies or practices using mobile lidar or photogrammetric data. This review highlights current trends, challenges, knowledge gaps, technical approaches, and performance measures. This review also discusses potential improvements or solutions relevant to the research problems.

The review commences with a brief background of the research problem as well as a summary of types of features to extract in this study. Next, it describes ODOT's current procedures and reference manuals for extracting and inventorying the features. The data collection section is then categorized into two sections, mobile lidar and photogrammetric data. In the mobile lidar section, the review describes mobile lidar system basics and a brief history of mobile lidar usage within Oregon DOT. In the photogrammetric data section, the review describes airborne data acquisition basics as well as some example data that can be potentially used for this project. Following this background, the review then covers critical concepts related to feature extraction methods categorized as pre-processing, segmentation, and classification. Finally, the review closes with a description of challenges and limitations of the state-of-the-art to guide this work.

2.1 FEATURE TYPES

2.1.1 Bike lanes

In Oregon, bike lanes (BL) are defined by a bike lane stencil and 8-inch stripe separating the bicyclist and motor vehicles (Figure 2-1a). Bike facility width is measured in feet from the face of the curb, the edge of pavement, or the center of the inside white line to the center of the 8-inch or outside stripe. Where the bike lane includes a street buffer, only measure the width where bicyclists ride.

Shoulder (SH) bikeways are designated with a 4-inch stripe and have a minimum width of 5 feet on a paved surface (Figure 2-1b). Shared Lanes (SL) are commonly used for low speed (25 mph or less) and low traffic streets (Figure 2-1c). No dedicated bike facility is designated; bicyclists and motor vehicles simply share the same travel lane. In these cases, shared lane markings (Figure 2-2) are optional to indicate bicycle positioning.

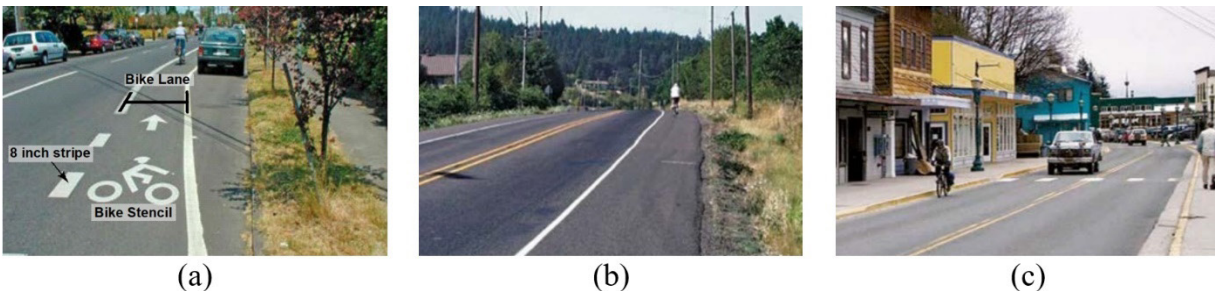


Figure 2.1: Types of bikeway: (a) bike lane with stencils and stripes; (b) shoulder bikeway; and (c) shared lane (ODOT, 2020)

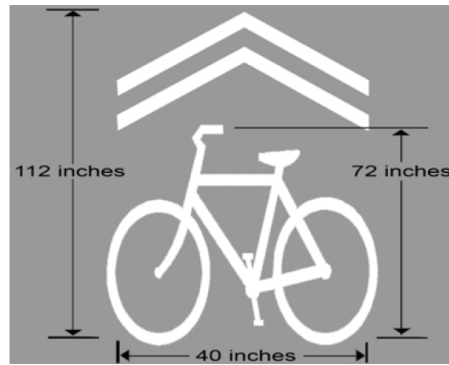


Figure 2-2: The shared lane marking (sharrow) symbol (ODOT, 2020)

Buffered bike lanes are conventional bike lanes paired with a designated buffer space separating the bicycle lane from the adjacent motor vehicle travel lane and/or parking lane. Buffered bike lanes provide greater shy distance between motor vehicles and bicyclists, thus improving safety. Buffered bike lanes are typically applied on streets with high travel speeds, high travel volumes, and/or high amounts of truck traffic (NACTO, 2022). Figure 2-3 provides some examples of buffered bike lanes recommended by the Federal Highway Administration (FHWA).

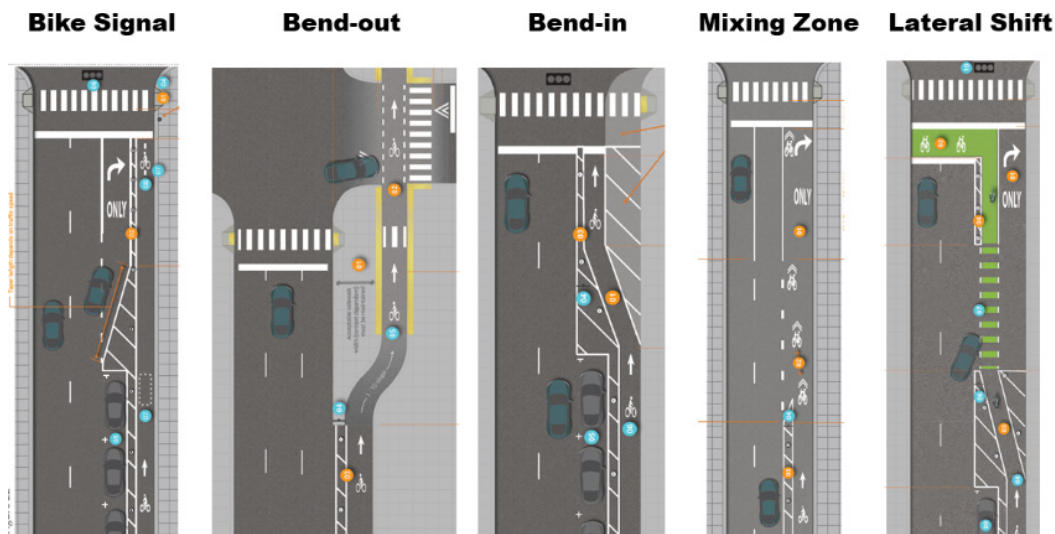


Figure 2-3: Types of buffered bike lanes (FHWA 2015)

2.1.2 Crosswalks

A crosswalk can be located at an intersection or midblock and must use white retroreflective markings applied in one of the patterns shown in Figure 2-4 with dimensional requirements show in Figure 2-5. An intersection is defined as a location where one road crosses paths with another (Figure 2-6a). A midblock crossing can be placed between intersections, often incorporating several design features to increase safety. Midblock crossings frequently include pedestrian islands (Figure 2-6b), which provide a safe refuge for pedestrians crossing two-way traffic. Pedestrians can check traffic one way, cross to the island, and then check traffic in the other direction before continuing to cross.

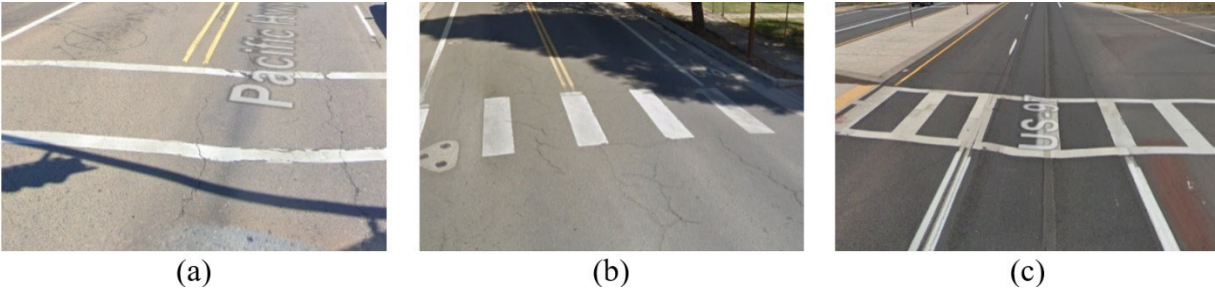


Figure 2-4: Types of crosswalk: (a) standard parallel; (b) staggered continental; and (c) ladder (ODOT, 2020)

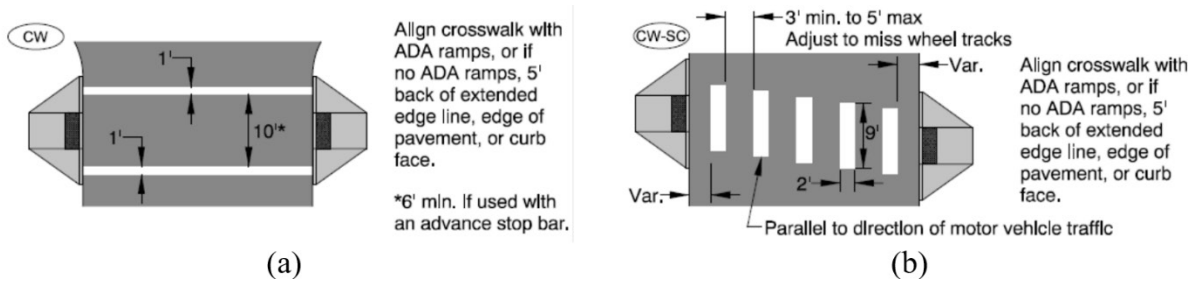


Figure 2-5: Patterns of crosswalk: (a) standard crosswalk and (b) staggered continental crosswalk (ODOT, 2020)



Figure 2-6: Locations of crosswalk: (a) intersection and (b) mid-block (ArcGIS Pro imagery)

2.1.3 Medians

A median is a reserved area that separates traffic moving in opposite directions. The median may simply consist of pavement; however, it is often used for a variety of purposes, such as decorative landscaping and trees. Figure 2-7 shows an example 3D point cloud of a median and

its associated profile view. Generally, the median will have a higher elevation than surrounding road surfaces, thereby detection of the curb enables separation of median.

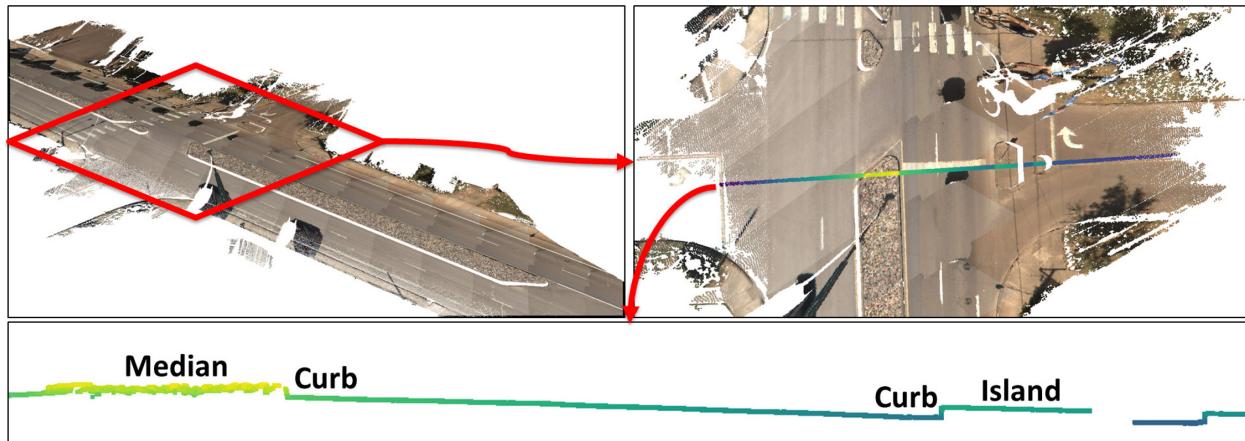


Figure 2-7: Example of gravel median in 3D point cloud data and its profile view

In TransGIS, ODOT classifies traffic barriers (which can serve as medians) as cable, concrete, and guardrail. Additional types of medians exist including depressed, raised, and closed, and painted or flushed (Iowa DOT, 2019). These may be paved and unpaved (e.g., gravel, landscaping, etc.) barriers.

2.2 OREGON DOT PROCEDURES

2.2.1 Inventory protocols

Currently, ODOT uses a variety of platforms to collect bike lane and pedestrian crosswalk features including aerial photos, street view, and boots on ground data collection efforts. These data are then used to manually tag features and add attributes in GIS. Construction contracts require the contractors to deliver these data products to ODOT. It is important to note that this is true for ODOT contracts only and does not necessarily include other agency updates at intersections. For existing assets, ODOT personnel (often interns) collect information across the vast highway network. Given that this is a manual process, issues can arise in the consistency and completeness of the provided data.

ODOT's inventory processes have changed over time and will continue to transform based on agency priorities and resource availability. The current process consists of data gathering with a mix of manual paper forms and digital processes. Digital data are acquired using a tablets, ruggedized laptops, or handheld devices during field work as well as extraction from video logs and other data sources on a desktop computer in an office. Regardless of the process used for inventory, preliminary work that includes identifying the segment of roadway to be inventoried, preparing data collection method (digital or physical spreadsheet), acquiring fundamental working knowledge of the digital video log (DVL) maintained by ODOT, and gaining knowledge of location and asset terminology is required. In addition, the inventory procedure requires ODOT staff to print a copy or have access to the *Highway Inventory Summary Report* for the relevant segment of the highway.

Inventory definitions for sidewalk facilities, marked crossings, shared use paths, parking facilities and curbs are included in the *ODOT Bicycle – Pedestrian Data Collection Guide* (2020) along with sample photographs. The guide also describes the features, attributes, and domain values for bike/sidewalk facilities. These definitions and photographs are used by ODOT staff to determine the appropriate details and codes for the asset during the data collection effort. For example, the location information for each traffic signal including its district number, highway number and suffix, roadway identifier code, overlap mileage code, begin / end mile points, and its position (left or right) needs to be documented by ODOT staff in the field to geographically reference and map the location of features along roadways. When doing desktop inventory in office, the location of a feature needs to be validated utilizing the tools such as DVL, straight line charts, FACS-STIP, and TransGIS that are available for that section of the roadway.

The inventory classifies bicycle facilities into three categories: 1) bike lane that includes an 8-inch stripe separating the bicyclist and motor vehicles and a bike lane stencil; 2) shoulder bikeway that is paved, minimum of 5 feet wide and have a 4-inch stripe; 3) shared lane where the speed limit is 25 mph or less. Shared lane markings are optional to indicate bicycle positioning. The condition of bike lanes is also documented in the inventory using one of the four ratings; good, fair, poor, blank (when there is not a bike facility at this location but is needed). There are three types of marked crosswalks, namely, standard parallel, staggered continental, and ladder that use white retroreflective crosswalk markings. A marked crosswalk can be located at an intersection or midblock. Crosswalk attributes collected for the inventory include whether the marked crosswalk is located at an intersection or midblock, first and second corners of the crosswalk, the image source (DVL or Google Earth), date, marking type, installation status, removal approval, and latest status check.

2.2.2 Linear Referencing Systems

Some assets in ODOT's inventories are tagged with geospatial coordinates from GPS/GNSS for precise positioning. Others are tagged using a Linear Referencing System (LRS). While much legacy data exists in LRS and the system can be intuitive to maintenance personnel in the field or to query information in a database, the LRS is not suitable to be the primary means of locating an asset in a geospatial database because it significantly degrades the positioning quality (Hallmark 2001) and is not compatible with modern sensors. Rather it should be relegated as a secondary attribute for positioning that is computed directly from the geospatial coordinates. In this way it can still be used for data inventory and interrogation purposes and all of the benefits of an LRS can be achieved. However, when it is used as the primary positioning information, several challenges arise:

1. Modern geospatial technology (e.g., GIS, GPS, lidar) work with rich geospatial coordinates and can pin assets down to a few centimeters (if desired and collected with appropriate technology—often meter level accuracy is sufficient). LRS provides referencing as a distance (and lateral offset) along a highway, this information can be sometimes stored at high accuracy but requires significantly more effort to obtain compared with geospatial coordinates readily provided with modern sensors.
2. As a result of this incompatibility with data acquired with modern sensors, the LRS system inhibits automation in the acquisition of information on highway assets and relies

primarily on manual data collection efforts, which are error prone, pose more safety hazards, are inefficient, and are costlier.

3. The LRS limits information from being able to be combined directly with other GIS layers and utilized in a variety of geospatial analyses. The bulk of statewide and local databases coordinated by many agencies are collected using geospatial coordinates.
4. The LRS is based on relative offsets and distances, which introduces many challenges in consistent positioning for long highway sections over decades given the dynamic landscape in Oregon (e.g., tectonics, landslides, and other earth movements). While such aspects will change the georeferenced position of an asset as well, that affect is minimized locally for that single asset compared with LRS where the physical length of the highway changes and all downstream assets are affected.
5. Machine learning is increasingly used to extract assets from datasets such as digital image logs and lidar. LRS alone does not provide sufficient positional information that it can be used for training/testing machine learning algorithms.
6. Geospatial coordinates can easily be converted to LRS values using GIS. However, converting from LRS to geospatial coordinates can be highly error prone and does not preserve the fidelity of the positioning information. As a result, the data quality is degraded.
7. Geospatial coordinates remain unaffected when a highway is realigned resulting in either longer or shorter distances along a path. In such cases, the LRS must be recomputed based on the changes in distances, which can cause confusion with legacy records. In some cases, there can be gaps in the milepost numbering rather than updating the entire length when a highway is shortened.
8. Along those same lines, linear distances are not the same from different vehicle lanes-which then poses challenges in field personnel navigating to features or to adequately reference assets tagged in the field using LRS.
9. GPS/GNSS and associated GIS map technology has become ubiquitous and available to personnel on many devices. Navigating via geospatial coordinates has become a simpler and less error prone process in many cases compared with using an LRS. In addition, GPS/GNSS is readily available on tablets and other devices used for data acquisition whereas LRS requires additional information to obtain.
10. While current personnel have been trained on using the LRS and can use it efficiently and effectively, future personnel will have “grown up” with using GPS/GNSS technology and require less training compared with using an LRS.

Notably, GPS can have challenges in dense urban or forested terrain. However, with the inclusion of additional networks in GNSS, positioning has significantly improved. As mentioned previously, LRS can still continue to be used with geospatial coordinates as it can be readily computed from a GPS/GNSS combined with GIS layers of the highway network. But when it is used by the primary

2.2.3 TransGIS

ODOT shares data internally and externally through the TransGIS platform, which provides access to a variety of datasets provided by many departments within ODOT. Amongst the many layers, TransGIS contains the following layers relevant to this project (Figure 2-8):

- **Marked Crosswalks** with attributes of ODOT district; milepoint; latitude and longitude; mid-block; crossing corners 1 and 2; high visibility marking; installation approval; last update year; and inspector name, crew, and agency. These include staggered continental (Figure 2-9) and standard parallel (Figure 2-10) crosswalks.
- **Sidewalks** with attributes of beginning and ending mileposts, side of road, surface type, buffer, width, physical condition, inspection year, and notes.
- **Bicycle Facilities** (Figure 2-11, categorized by Bike Lane, Shoulder > 5', and shared lane). Contains attributes of facility type; beginning and ending milepost; side of road; needed; width (ft); physical condition; notes, and inspection year.
- **Traffic Barriers** (Figure 2-12, categorized by Cable, Concrete, and Guard Rail). Contains attributes of highway name; highway number (and suffix); roadway ID; mileage type; overlap mileage code; beginning and ending milepost; barrier description; side of road; begin terminal type; height; condition; and shared; barrier type, height, and condition; end terminal type, height, condition, and shared; concrete barrier construction and connection type; rail post spacing and type; block out; last inventory year; and comments.
- **ADA Ramps:** Rated as Good, fair, poor, missing, and not needed. Contains attributes of: milepoint, cross street name, corner position, corner type description, ramp position, ramp need status, ramp style, ramp functional condition, reason not compliant, inspection year.

All datasets also have an LRS key, effective date, and GIS process date.

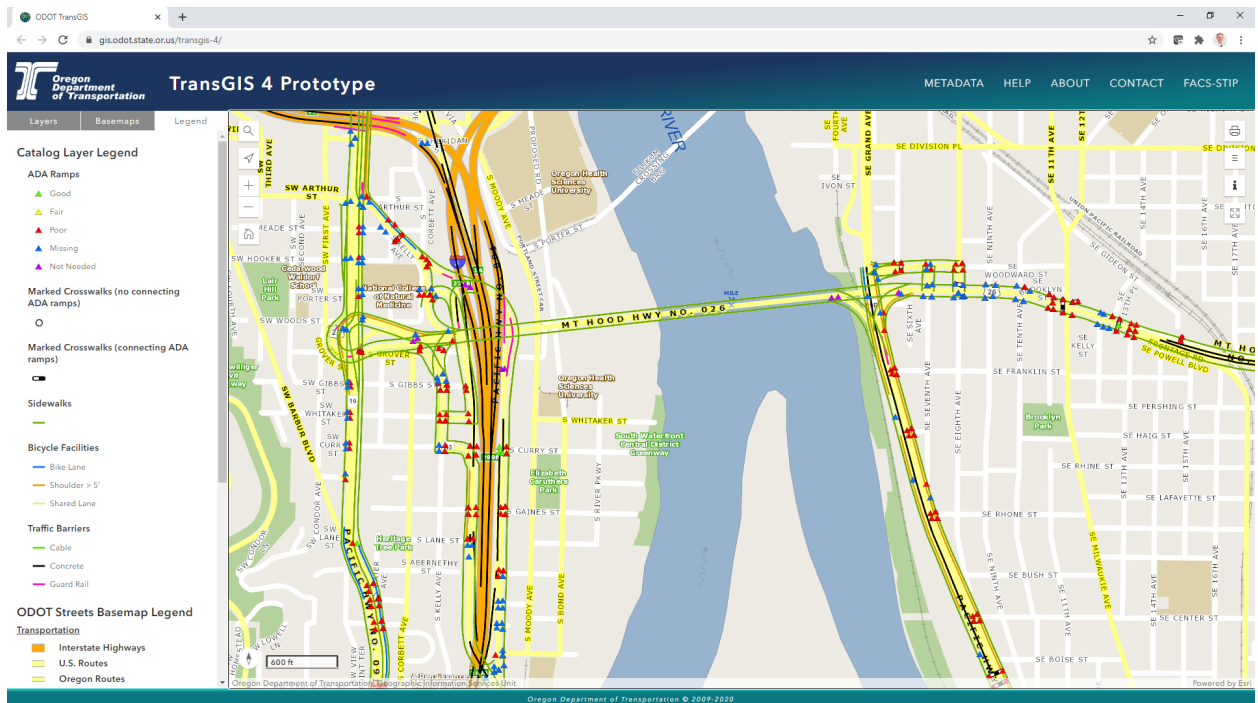


Figure 2-8: Overview of example datasets in TransGIS relevant to this study.

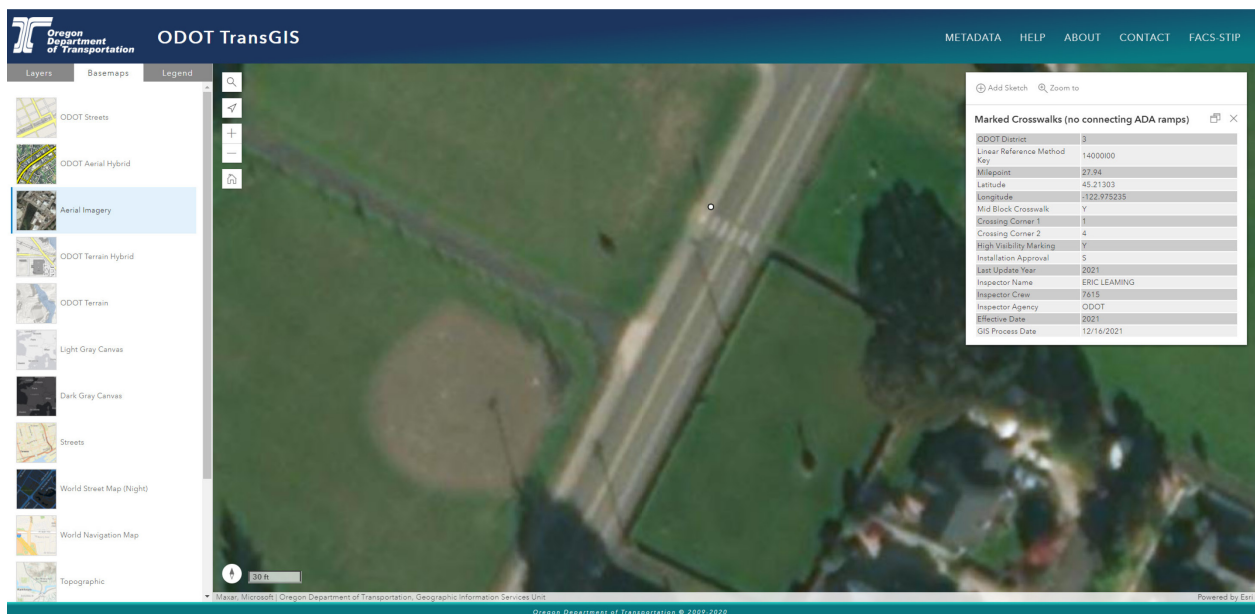


Figure 2-9: Example staggered continental crosswalk in TransGIS.



Figure 2-10: Example standard parallel crosswalk in TransGIS.

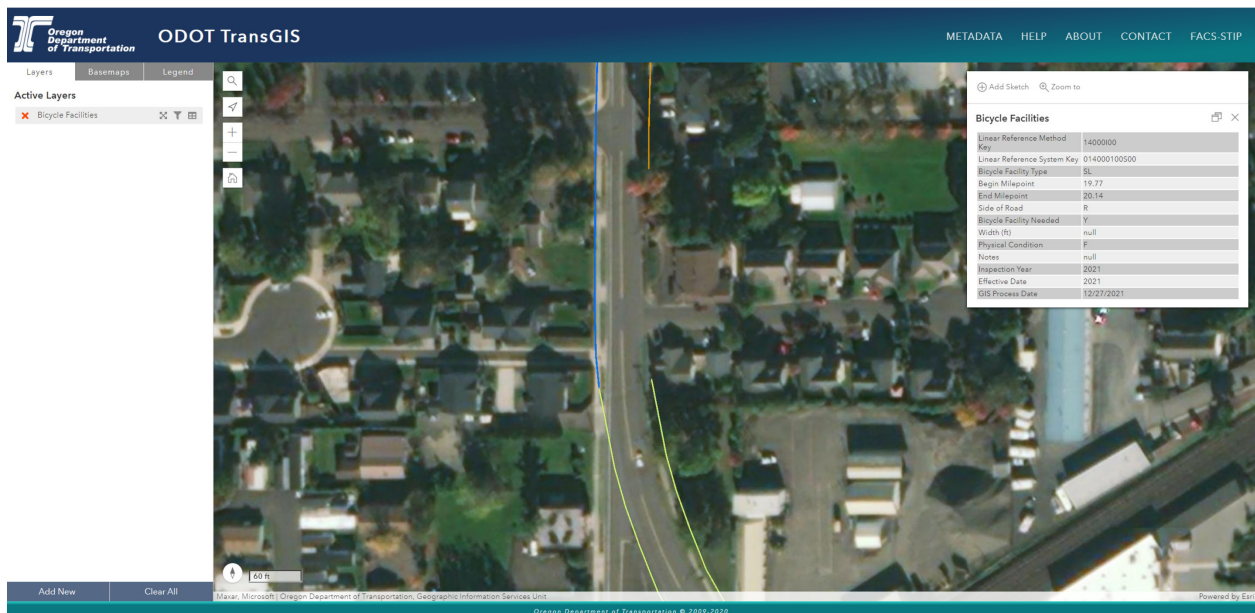


Figure 2-11: Example bike lane in TransGIS



Figure 2-12: Example traffic barriers layer in TransGIS

2.3 MOBILE LIDAR TECHNOLOGY

2.3.1 Mobile Lidar System Applications

Mobile lidar (ML, also called Mobile Laser Scanning, MLS, or Mobile Terrestrial Laser Scanning, MTLs) systems acquire detailed 3D data efficiently from a moving vehicle at highway speeds. The rich information captured with lidar provides several benefits and, as a result, is being widely adopted by DOTs across the country (*NCHRP Report 748 and NCHRP Synthesis 446*). A key benefit of lidar is that the same dataset can be exploited by multiple personnel for a wide variety of applications (Figure 2-13), minimizing the need for multiple data collection efforts. This versatility has led to common usage of the phrase, “Collect once, use many times” regarding lidar.

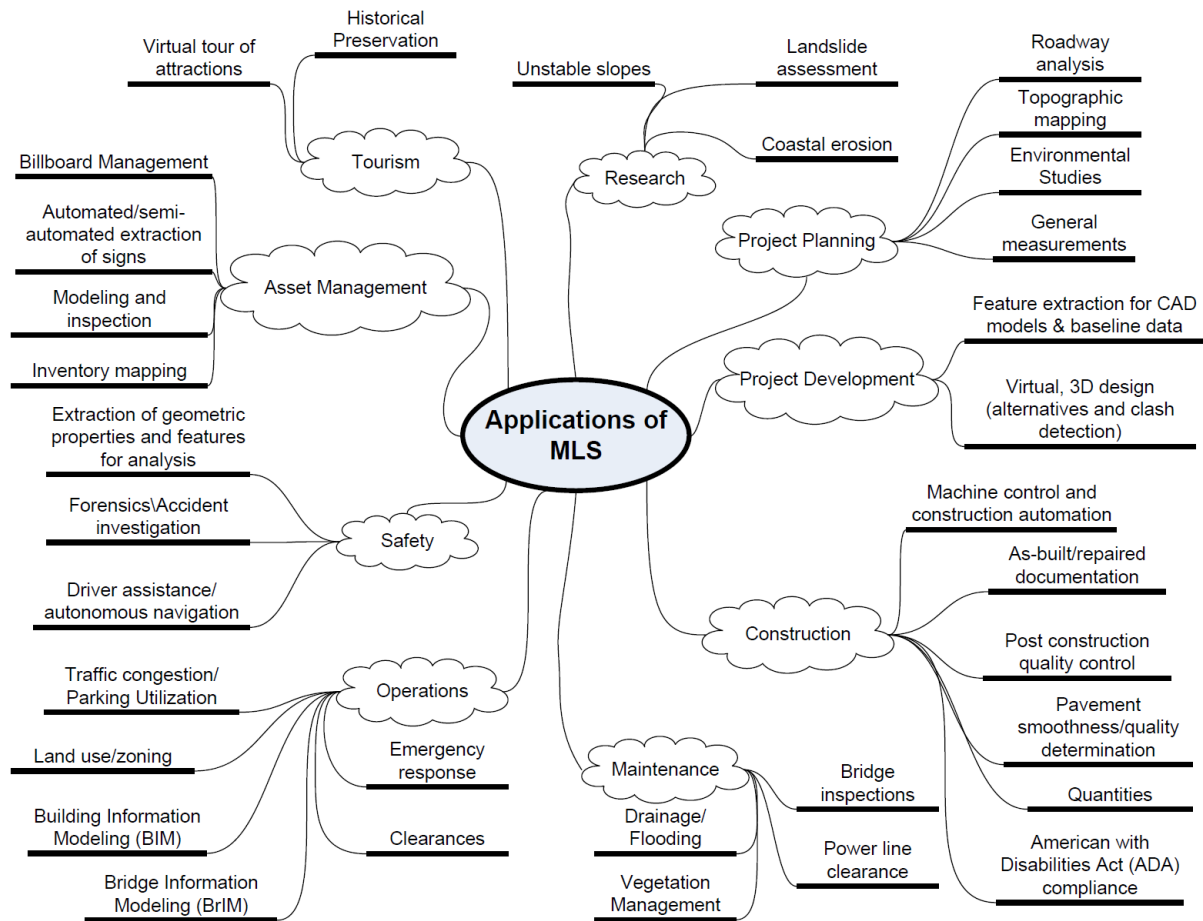


Figure 2-13: Sample applications using mobile lidar technology in transportation. From NCHRP Report 748 (Olsen et al. 2013)

With mobile lidar, one can remotely survey a site in relative safe conditions, minimizing the danger to field crews and the travelling public. Lidar also enables a much more efficient and thorough field survey, minimizing the need for costly repeat visits to a site to obtain information.

The comprehensive information provided by lidar greatly improves the detail in models used throughout the design process and hence, reduces uncertainty in decision making. The additional information that is resolvable in lidar data enables topography and other features to be modeled at a higher level of detail and accuracy over traditional techniques. This detailed, 3D, virtual world provides personnel in the transportation agency with a much better understanding of the field conditions and variability throughout a highway corridor.

2.3.2 Mobile Lidar System Basic Technical Information

A wide range of ML systems exist depending on the scope and purpose of the data collection effort. Puente et al. (2013) describes and compares configurations of a variety of systems. They can be designed to be specialized for certain applications such as pavement analysis or configured for general data acquisition. Lower cost asset management & mapping systems (~\$2-

400k) can achieve sub-meter (<3.28 ft) accuracies at the network level and decimeter (several inches) accuracies at the local level. Survey grade systems (~\$1 million) can achieve centimeter (~1in) level accuracies at both the network and local level. While highest accuracy has required the use of dense targets, higher accuracy and more reliable results can be obtained by performing multiple passes of a section, enabling improved verification of GNSS quality as well as trajectory enhancements by averaging multiple passes (Nolan et al. 2015a; 2015b).

Another key benefit of mobile lidar is the ability to integrate other sensors onto a single mobile platform (NCHRP Synthesis 446). This enables the collection of a wide variety of important metrics needed for various applications from a single data collection effort.

Mobile lidar provides rich data to support asset management. Georeferencing can be completed directly with the combination of components included on the scanner; however, for highest accuracy applications, rigorous survey control points are often established. Most mobile lidar systems co-acquire images from panoramic cameras, enabling RGB color to be mapped to the point cloud. The cameras are calibrated such that the images are linked to the mobile lidar data. Hence, features can be extracted from the imagery and then mapped to the point cloud in some software.

In addition to a wealth of geometric and color information across the roadway and surrounding area, mobile lidar data also records return signal intensity values (Kashani et al. 2015), which contain information on surface characteristics, including reflectance (Figure 2-14). As a result, intensity information can be very useful in extraction of features (e.g., signs, lane markings, etc.) However, these intensity values are affected by many factors including system parameters and laser characteristics (e.g., wavelength, transmitted laser power, beam divergence, receiver aperture, system transmittance), acquisition geometry (e.g., range, incidence angle) and other environmental variables (e.g., atmospheric transmittance, presence of moisture on the surface).

Kashani et al. (2015) provide a comprehensive overview of intensity, applications, and different types of radiometric calibration and correction routines. These approaches can be classified in the following general processing levels:

- Level 0: Raw intensity (no correction) – values directly provided by the system.
- Level 1: Intensity correction – corrections applied for factors such as range and angle of incidence.
- Level 2: Intensity normalization – intensity values are adjusted to provide relative consistency between passes.
- Level 3: Rigorous radiometric calibration – intensity values are corrected based on theoretical and/or empirical corrections to standard reference measurements, providing consistency between lidar datasets and other datasets.

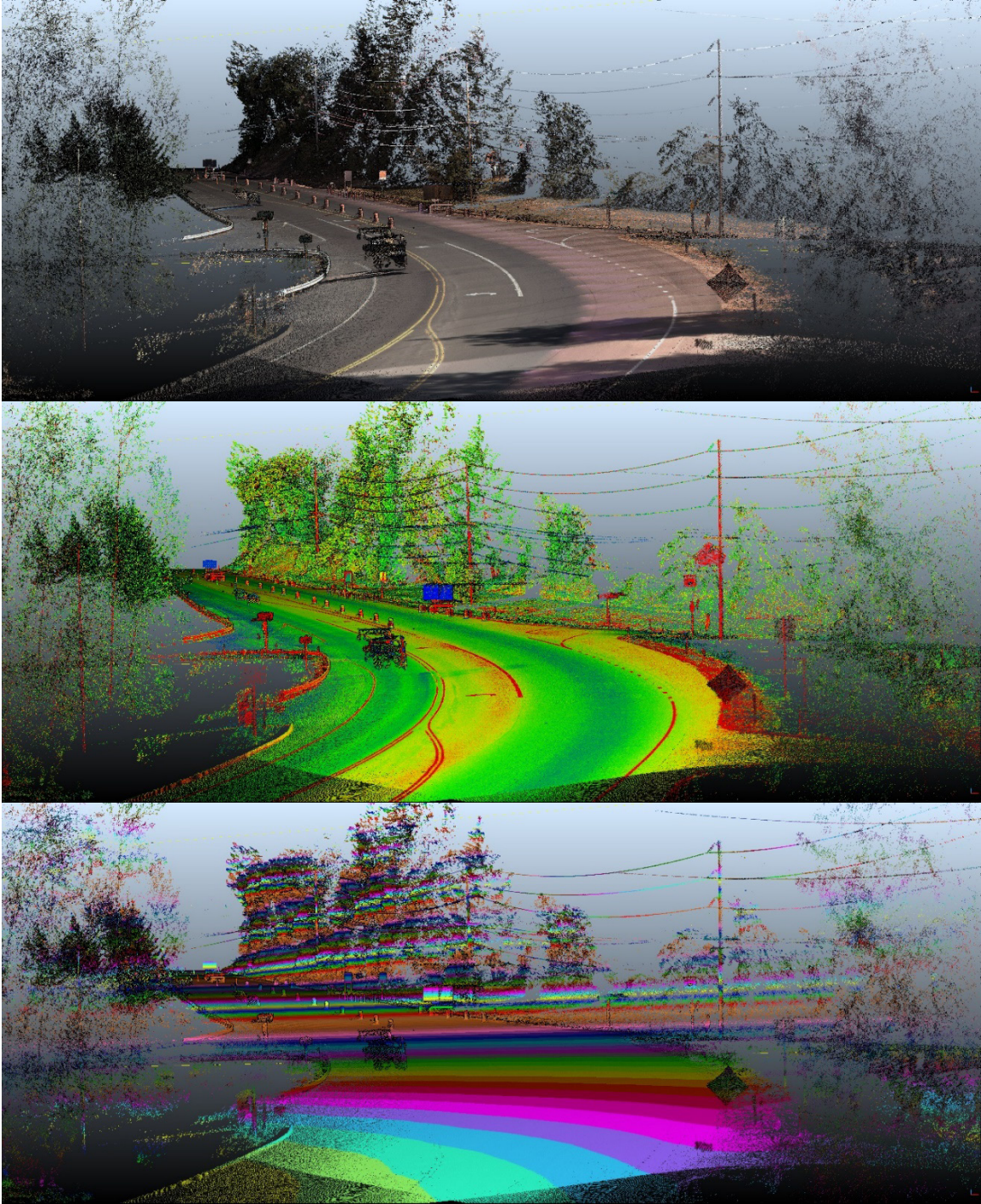


Figure 2-14: Mobile lidar data collected by ODOT on Highway OR-231 at the Spangler Landslide, near Molalla, OR. In the top image, the lidar data are colored using the imagery collected from the camera integrated in the mobile lidar system. In the middle image, the same point cloud is shown displayed by intensity. Although the intensities shown here are un-calibrated, pavement markings and other features are clearly distinguishable. The bottom image shows the point cloud color-coded by elevation, where colors are alternated every 0.20 m increase in elevation.

2.3.3 Mobile Lidar at Oregon DOT

Oregon DOT has been an early adopter of ML technology. When the technology first became available, Oregon DOT contracted its use on several highway projects. In 2011, Oregon DOT purchased a TopCon IPS2 ML system primarily for asset management purposes as well as to expand expertise with ML technology. To the authors' knowledge, Oregon DOT was the first state DOT to own a ML system. Data from mobile lidar surveys were used to extract features to update asset management databases. As more personnel within Oregon DOT utilize the data, additional applications continue to be identified.

After several years working with and becoming more comfortable with ML technology, ODOT purchased a survey-grade system, the Leica Pegasus:Two (Figure 2-15). The higher data quality enables it to be used for a broader range of applications. This versatile system can also be mounted on other vehicles, ATVs, or boats to acquire data in more difficult to reach locales. The system includes 8 cameras (7 providing a panoramic view and one focused on the pavement), capturing a detailed video log of the highway in addition to the geometric information.



Figure 2-15: Oregon DOT's current mobile lidar system, Leica Pegasus:Two

Oregon DOT's Engineering Technology Advancement (ETA) group explores additional opportunities to utilize ML data to either complement or replace current data collection procedures. These activities are described on their webpage: (<https://www.oregon.gov/ODOT/HWY/ETA/Pages/Mobile-LiDAR-Applicaitons.aspx>). Oregon DOT's mobile lidar system is being utilized for acquiring survey data for project development, measuring vertical clearances, measure widths of drive approaches, asset management, pavement evaluation, slope stability monitoring, accident reconstruction, and many more.

As an example, the State of Oregon raised the speed limit on highways in central and eastern Oregon. This project required ODOT to evaluate sight distance as well as passing zone striping with the increased speed to ensure that highways would still be compliant with safety regulations at these higher speeds. Mobile lidar data already collected from ODOT's routine mapping efforts were extensively used in the supporting passing distance studies and ensured that ODOT could complete the necessary remediation quickly to meet strict timelines.

Recently, Oregon DOT evaluated the usage of lidar for evaluating pavement markings (Olsen et al. 2019a, SPR 799). The study conducted a rigorous radiometric calibration of ODOT's mobile lidar system and found that mobile lidar was an effective tool to evaluate the condition of most road markings, particularly those close to failure. Very reflective markings; however, could not be properly read as the signal saturated. This research project resulted in the development of a new tool, called ROME for extracting and evaluating pavement markings. ODOT has continued to use this tool to extract pavement markings and perform evaluations for different projects as well as in regions throughout the state to build up an asset inventory database. Currently Oregon DOT is conducting a pilot project to use these extracted markings to help guide snow plows in precarious canyons.

In another recent study (Olsen et al. 2020, SPR809), Oregon DOT explored the usage of mobile lidar to evaluate rockfall activity on cut and natural slopes near highways. The study found that mobile lidar could provide useful information to monitor the slopes in some cases; however, the system did have difficulty in capturing the upper portions of steep slopes given the poor scanning geometry. Additionally, multiple passes or slower speeds were required to capture data at adequate resolution.

ODOT continues to make great strides in its vision is to utilize mobile lidar and other technologies to provide a real-time, digital transportation system or virtual corridor (Singh et al. 2009). When construction projects or maintenance are completed, the data would be updated to reflect those changes.

2.3.4 Mobile Lidar Standards, Specifications and Guidelines

One of the first DOTs to develop formal ML specifications was Caltrans (Chapter 15 California Survey Manual). These specifications have been modified and utilized by other DOTs such as Florida DOT. Caltrans has continued to develop best practices, workflow, and training documentation for mobile laser scanning data collection (Yen et al. 2014).

NCHRP Report 748 provides performance-based guidelines for the use of mobile lidar in transportation applications (Olsen et al. 2013a). Based on interviews with state DOTs and service providers, the report indicates that transportation agencies have a strong interest in mobile lidar going forward, but there are very few examples of best practices and/or in-depth discussions of results. This guideline establishes nine data collection categories (DCC) that are appropriate for the specific transportation applications based on resolution and accuracy requirements. The guidelines also provide general recommendations concerning the critical issue of data management. It is divided into two main sections: Management and Technical. The management portion contains a discussion of applications, workflows, data mining, procurement process, decision making, an implementation plan, and currently available guidelines. The

technical section describes the components of MTLs, error sources, calibration and correction, accuracy and resolution requirements and specification, quality control methods, considerations for common applications, information management, deliverable specification, and future trends. Appendices also contain sample calibration reports and templates for developing scopes of work. This work was developed into an e-learning website (<http://learnmobilelidar.com>), which includes online, interactive learning modules, a detailed and searchable reference list, and user forums to help educate about mobile lidar usage to support transportation applications.

2.4 PHOTGRAMMETRIC DATA

2.4.1 Photogrammetry Basics

Photogrammetry is broadly defined as the process of deriving information through measurements made on photographs or digital images (e.g., Mikhail et al., 2001). The primary objective in most photogrammetric projects and tasks is to perform quantitative measurements from overlapping sets of imagery acquired from aerial cameras, generally installed on airplanes (Figure 2-16), helicopters, or un-crewed aircraft systems (UAS). Imagery can alternately be acquired from a nearly unlimited range of other platforms, including vehicles, boats, or satellites, to list just a few of the more common examples. Photogrammetry is occasionally characterized as a subfield of remote sensing, the field dedicated to obtaining information about objects or phenomena at a distance, usually through sensing reflected or emitted electromagnetic energy. However, due to the emphasis on 3D spatial information and precise coordinates, photogrammetry is more typically considered its own distinct field, albeit one which overlaps with remote sensing and has similarities to a number of related fields and technologies, such as lidar.

Photogrammetry has the interesting characteristic of being both a mature and an emerging discipline. The field traces its roots to at least as far back as the mid-1800s, with its mathematical foundations developed well before that (Stirling, 1990; McGlone et al., 2004; Wolf et al., 2014), yet advances in feature matching, and new forms of photogrammetry leveraging new algorithms from the field of computer vision continue to lead to new innovations in photogrammetry. Among the notable recent advances is the widespread use of structure from motion/multi-view stereo (SfM/MVS) photogrammetry, often applied to imagery acquired from low-flying UAS (Westoby et al., 2012; Tonkin et al., 2014; Eltner and Sofia, 2020).

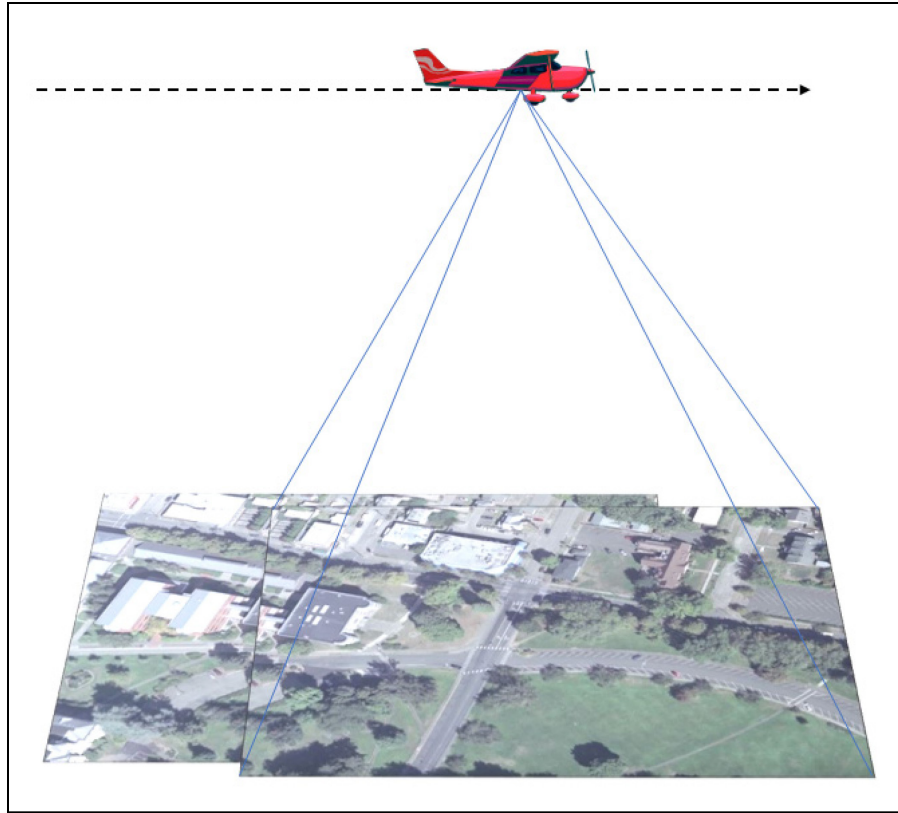


Figure 2-16: Stereo photogrammetry: aerial imagery acquisition schematic, showing a pair of overlapping images of the OSU campus.

2.4.2 Road Extraction via Photogrammetry

Similar to the field of photogrammetry itself, the specific focus area of object detection from imagery is both a well-studied research topic, and also an area of rapid ongoing development. Substantial research has focused on extraction of linear features, and roads in particular, due to the importance to transportation agencies (Quackenbush, 2004). Early methods of feature extraction were primarily manual, often requiring a human operator to digitize the features in imagery, although so-called “semi-automated” approaches, in which computer hardware and software aided the human analyst grew increasingly prevalent in the 1980s-1990s (Pigeon et al., 1999). Early forays into automation focused on conventional spectral clustering techniques, as well as edge detection and template matching (e.g., Taniguchi and Kawaguchi, 1989; Benjamin and Gaydos, 1990; Gruen et al., 1995; Baumgartner et al., 1999). Song and Civco (2004) investigated a support vector machine (SVM) for road extraction. Not surprisingly, many recent approaches to road extraction from imagery utilize deep learning (e.g., Buslaev et al., 2018; Xu et al., 2018).

Of highest relevance to this project are photogrammetry or imagery-based approaches to feature extraction that focus not merely on road networks, but on specific features, such as lane markings, bike lanes, and crosswalks. Kim et al. (2006) investigated a set of algorithms for efficient extraction of road pavement markings from aerial imagery. Jin et al. (2009) investigated extraction of road lanes from stereo aerial imagery using maximum likelihood segmentation.

Some noteworthy approaches rely not only on imagery alone, but also other forms of data (e.g., lidar) or existing maps or GIS databases for road feature extraction. Ishino and Saji (2008) presented a method of extracting crosswalks and other road markings utilizing morphological image processing using both aerial imagery and existing digital maps. Kasemsuppakorn and Karimi (2013) investigated extraction of crosswalks (as well as other “pedestrian path types,” such as sidewalks) from aerial imagery and lidar.

2.5 OBJECT EXTRACTION METHODS

2.5.1 Pre-processing

2.5.1.1 Ground filtering

Ground filtering is a common process to separate ground (e.g., terrain, road surface, sidewalk) and non-ground (e.g., buildings, poles, trees) points, which usually serves as a pre-processing step for more advanced feature extraction and analysis. By classifying the ground points, the data volume needing to be analyzed can be significantly reduced such that subsequent processes (e.g., segmentation, feature extraction, object recognition) can be less computationally complex. Numerous ground filtering methods have been developed for processing airborne laser scanning (ALS) data. Sithole & Vosselman (2004) evaluated a number of ALS ground filtering methods that can be categorized into slope-based, block-minimum, surface-based, and clustering/ segmentation. Then Meng et al. (2010) performed a comprehensive review on additional ground filters and classified them into segment/cluster, morphology, directional scanning, contour, TIN (triangular irregular network), and interpolation. More recently, with a similar framework, Chen et al. (2017) reviewed more recent ground filtering methods for ALS data. It is worth noting that the outcome of ALS ground filtering can be different depending on the target applications or products. For those who focus more on the topographic mapping, some ground filters directly produce a digital terrain model (DTM) in forms of 2D grids/rasters (e.g., digital elevation model (DEM)) or triangular mesh (e.g., triangular irregular network (TIN)). To make these approaches work for more advanced feature extraction and analysis, the outcome terrain model needs to be mapped back to the input point cloud before further processing. In contrast, the methods that directly classify the point cloud as ground and non-ground are more suited for extracting various features.

The point cloud data collected from a ground-based lidar system (e.g., terrestrial laser scanning (TLS), mobile laser scanning (MLS)) often requires modification of the ALS ground filtering or different processing frameworks because comparing against ALS data, TLS and MLS usually feature higher resolution and accuracy as well as more details captured on the side of the object. One common approach to organize the point cloud for identifying ground is to rasterize the 3D point cloud into a 2D image where each pixel can store representative information from the points located within the range of it for computing various characteristics such as point density, elevation, elevation variation, and so on (Hernández and Marcotegui (2009), Serna and Marcotegui (2014), Yang et al. (2012)). Another strategy to organize the point cloud is to divide the MLS data into 2D vertical profiles (Wu et al., 2016. Teo and Yu, 2015). Although projecting 3D data to either horizontal or vertical plane can reduce the computation complexity and thus

increase the efficiency, such approaches can be limited by the information loss caused by the projection process. As a result, other methods developed methods that directly work in 3D space and were able to utilize richer information to detect ground points (e.g., Ibrahim & Lichti, 2012; Lin and Zhang, 2015; Yadav et al., 2017; Husain & Vaishya, 2016). Table 2-1 summarizes these approaches.

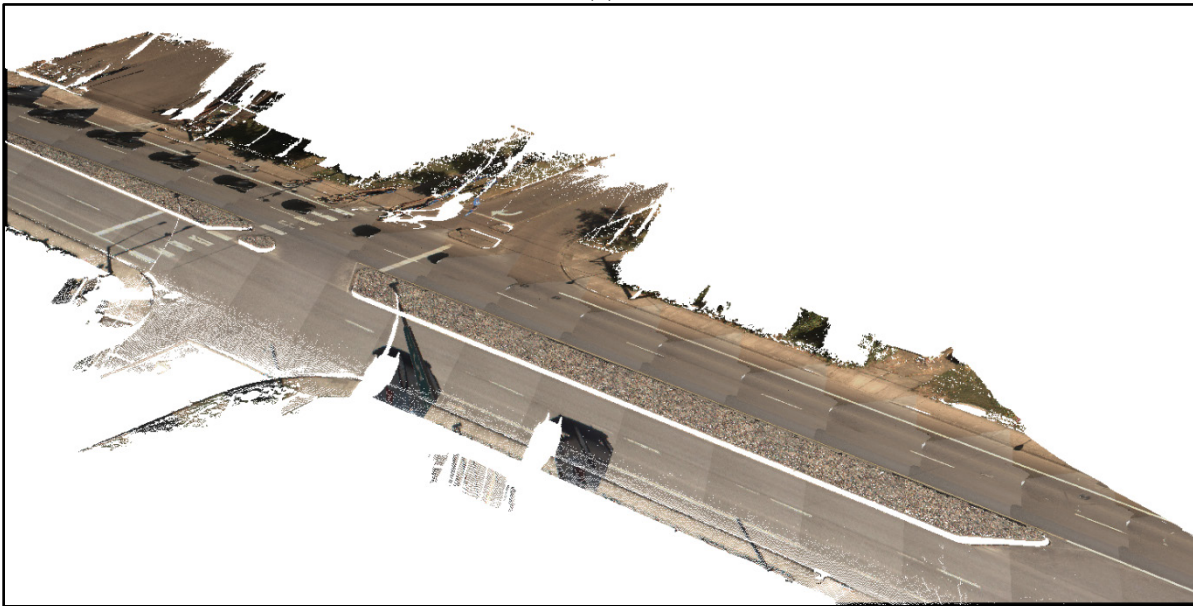
Table 2-1: Summary of studies in ground extraction from point clouds

	Methods	Methods	Methods	Characteristics	Characteristics	Characteristics
Study Name	Raster	3D-based	Scan line	Point density	Elevation variance	Elevation jump
Yang et al. 2012	✓	-	-	-	✓	-
Hernández and Marcotegui, 2009; Serna and Marcotegui, 2004; 2014	✓	-	-	-	-	✓
Wu et al. 2016	✓	-	✓	-	✓	-
Ibrahim and Lichti et al. 2012	-	✓	-	✓	-	-
Husain and Vaishya, 2016; Yadav et al. 2017	-	✓	-	-	✓	-
Lin and Zhang, 2015	-	✓	-	-	-	✓
Teo and Yu, 2015	-	-	✓	-	-	✓

In the research team's prior work, a number of ground filters have been developed focusing on different scenarios (Che and Olsen, 2017, Che et al., 2021a, Che et al., 2021b). For example, Che and Olsen (2017) proposed a very fast ground filtering approach for TLS data based on scanline density analysis where the point density at each point in a 2D window is normalized considering the scan geometry. Then, Che et al. (2021a) developed a road extraction method including a ground filtering approach based on the segmentation results for MLS data. In this MLS ground filtering, scanlines were extracted by reconstructing the scanning acquisition structure and utilized to identify ground segments. Although these two approaches are shown to be efficient and accurate in terms of classifying ground points, they are limited by their requirement of the input data. They were designed to work with organized TLS or MLS data with known scan pattern information. Although the research group also developed a trajectory reconstruction approach to reconstruct the unorganized MLS data into the acquisition pattern, the ground filtering can only cope with a single sensor and pass each time during the scanline analysis. To tackle these limitations, the research team recently developed a more generalized ground filtering approach (Che et al., 2021b) that can cope with any unorganized point cloud data disregarding the data acquisition techniques (e.g., TLS, MLS, ALS, SfM/MVS). The approach implements a new voxelization technique that mitigates loss of geometric details to perform multi-scale analysis to classify ground and non-ground surfaces. The accuracy, efficiency, scalability, and versatility were demonstrated qualitatively and quantitatively with a number of large-volume datasets. The research team also tested it on a number of MLS datasets collected by ODOT with Leica Pegasus: Two system (Figure 2-17) where it has produced satisfactory results.



(a)



(b)

Figure 2-17: Example of *Vo-SmoG* ground filtering on a mobile lidar data collected by ODOT with the Leica Pegasus:Two system: (a) Original and (b) Ground points

2.5.1.2 Road surface extraction

After filtering the non-ground objects, the remaining ground points can be separated further into road or non-road surfaces. This significantly reduces the area evaluated to extract road markings, thus reducing false predictions and excess computation time. Table 2-2 summarizes common approaches found in the literature. Broadly, they can be divided into point-based approaches that use mobile lidar point cloud data and image-based approaches that use aerial or satellite imagery.

Table 2-2: Summary of road surface extraction methods

	Method s	Method s	Method s	Method s	Characteristic s	Characteristic s
Study	Raster	3D	Scanline	Image- based	Spectral values	Geometry
Serna and Marcotegui, 2013	✓	-	-		-	✓
Kumar et al. 2013	✓	-	-		✓	✓
Rodríguez-Cuenca et al. 2015	✓	✓	-		-	
El-Halawany et al. 2011, Rodríguez-Cuenca et al. 2016	✓	✓	-		-	✓

Study	Methods	Methods	Methods	Methods	Characteristics	Characteristics
	Raster	3D	Scanline	Image-based	Spectral values	Geometry
Ibrahim and Lichti, 2011, Miraliakbari et al. 2015, Xu et al. 2017, Zai et al. 2018	-	✓	-		-	✓
Yadav et al. 2017	-	✓	-		✓	✓
Miyazaki et al. 2014, Cabo et al. 2016	-	-	✓		-	✓
Maboudi et al. 2018 Tao et al. 2019	-	-	-	✓	✓	✓

Many point-based studies identify the road boundaries (e.g., curbs, barriers) to separate the road surface from the non-road surface (e.g., grass, sidewalk). It can be commonly assumed that the road surface has a lower height than non-road surfaces, thereby detecting a height jump (Figure 2-18) enables road boundary extraction. In general, the detection of the height jump can be categorized into: (1) rasterization, (2) 3D-based, and (3) scanline methods. In rasterization methods, the point clouds are organized into 2D grid cells using the height values. Often, minimum height values are stored in the raster image because road surface usually contains the lowest points on each grid. Subsequently, the regions with drastic height changes can be detected as road boundary.

For the boundary detection, several algorithms have been proposed, such as geodesic elongation algorithm (Serna and Marcotegui, 2013), Canny edge detection (El-Halawany et al. 2011), morphological operation (Rodríguez-Cuenca et al. 2015; 2016), snake model (Kumar et al. 2013), many of which benefit from existing image processing libraries. To improve boundary detection, Kumar et al. proposed to generate multiple feature images using reflectance and pulse width, which are proved to be an effective means to extract road surface from rural roads without a well-defined (e.g., sufficient height or slope difference) road boundary. 3D-based methods can be computationally more expensive, but often advantageous to preserve details, especially compared to the rasterization methods. Non-ground objects with a low point density can be detected and removed based on the fact that they are usually distant from the mobile lidar system's trajectory. Several methods have been proposed to detect the road boundary from the point clouds, such as a Gaussian function (Ibrahim and Lichti, 2012), region growing (Miraliakbari et al. 2015), and graph-cut energy minimization (Zai et al. 2017). Xu et al. (2016) and Zai et al. (2017) organized the point clouds into 3D voxels to detect curbs. Yadav et al. (2017) organizes 3D points into 2D grid cells, then filter out points with high intensity variability (e.g., grass-soil surfaces) from road surface. After boundary extraction, splines can be fitted to refine the boundary (Miraliakbari et al. 2015 and Yadav et al. 2017). In scanline approaches, angle or time information can be used to transform the point cloud into a set of scanlines. This can potentially reduce the computational complexity of the road boundary extraction while retaining the details. Miyazaki et al. 2015 perform a line-based region growing technique to discriminate the curbs from other regions. Cabo, et al. 2016 use the Douglas-Peucker algorithm to simplify the scanlines into a set of straight lines for further filtering.



Figure 2-18: Example of the height jump along the curb, serving as a road boundary in point cloud data.

In image-based approaches, early studies detected the centerlines of roads, but recent studies focus on segmentation of all road pixels from images. Overall, image-based road surface extraction methods can be categorized into feature-based and object-based approaches (Tao et al. 2019). Feature-based methods use spatial features (e.g., line, edge, and ridge) and spectral features (e.g., RGB and NIR) to extract road surfaces. Unlike mobile lidar data, aerial or satellite images can provide multi-spectral information, such that ratios images can be generated and used for segmentation. For example, vegetation can be readily detected and separated using normalized difference vegetation index (NDVI). Moreover, given the fact that roads are mostly elongated objects with linear properties, prior knowledge (direction and width) can also be used to detect and refine roads. Object-based approaches are more preferable for road extraction from complex and noisy images, such as urban areas, because they are less sensitive to salt-and-pepper errors. Object-based approaches partitions an image into non-overlapping regions such that each region is homogenous according to some coherent features whereas the union of each two adjacent regions is not homogenous (Weng, 2012). Edge detection or region growing methods are commonly adopted for object segmentation. Once objects are segmented, road objects are separated from others using a rule-based or machine learning classification. Recently, convolution neural networks (CNN)-based methods are receiving great attention because it enables fusing multiscale information and combining spectral and spatial features (Tao et al. 2019).

Image-based approaches also require homogeneous radiometric conditions (e.g., reflectance) for the object segmentation because radiometric discrepancies on the same object would increase intra-class spectral variance, resulting in fragmented segments (Weng, 2012). In order to smooth the image while retaining the road edge, Maboudi et al. (2018) applied the guided filtering. Tao et al. (2019) proposed a well-designed spatial information inference structure (SIIS) to capture and transmit the road-specific contextual

information. Unlike typical CNNs that transfer information layer by layer, SIIS splits the feature map into chunks along row or column and applies the information processing units sequentially. In this way, information can be roughly propagated to connect different segments on the same road.

2.5.2 Segmentation

2.5.2.1 General

Point cloud segmentation groups points based on common attributes such as geometry, radiometry (intensity), and/or RGB color. This process can turn millions of discrete points into a handful of clusters, which can substantially reduce the data volume. Subsequent processes can be performed with each segment as basic elements rather than operate on each individual point. Moreover, because each segment contains a number of points, more and richer information can be extracted from these points given a larger sample size. This is particularly important for feature extraction and classification of point cloud data because the discrete point cloud does not inherently have semantic information and can be seen as independent observations to each other. Furthermore, by reducing the data volume via segmentation, more complex analysis and computation can be applied to the point cloud data for more advanced applications. A lot of software packages support point cloud segmentation to some degree. However, most require manual delineation and other manual processes to isolate the segments, which can be very tedious and time-consuming. While it is still an active field of research, much work has been done to address the topic of automatic point cloud segmentation.

For ALS data, most methods focused on segmenting building roofs in an urban area due to the limited resolution and viewing angle. Because the vast majority of the building roofs consist of planar surfaces, the task of point cloud segmentation for ALS data has substantial overlap with the task of extracting and modeling planar surfaces. Xia et al. (2020) provided a thorough literature review that summarized the state of the art in extracting geometric primitives from 3D point clouds. More recently, a few new algorithms have been proposed targeting roof segmentation (e.g., Poz et al., 2020, Wang et al., 2020, Zhu et al., 2021, Zhang et al., 2021). Common techniques for plane fitting include the Hough transform, principal component analysis (PCA), random sample consensus (RANSAC), and region growing, which are all proved to be effective by numerous researchers (e.g., Zhao et al., 2021). ALS also has a wide variety of applications in forestry given its advantage of being an active remote sensing technique, which has the capability of penetrating the tree canopies comparing against other passive sensors (e.g., photogrammetry, satellite imagery). Segmentation within a forest scene is usually focused on individual tree segmentation (e.g., Wang et al., 2019, Yang et al., 2020). These segmentation approaches should be applicable to point clouds generated from SfM or photogrammetry or UAS-lidar systems given the similar view angle. However, very few works have been done such evaluation rigorously considering the different noise levels of these systems as well as the variability of the resolution and uncertainty within a dataset.

Similar to the aforementioned techniques in segmenting ALS data, some approaches are developed for ground-based lidar data (e.g., TLS, MLS) to extract planar surfaces or certain features from buildings (interior and/or exterior), bridges, and other anthropogenic objects to support a specific application (Wang and Kim, 2019). For example, Maalek et al. (2018) utilized PCA to extract and segment linear and planar features at a construction site to track progress. For generic segmentation methods, voxelization and super-voxelization techniques have been widely used (e.g., Vo et al., 2015, Lin et al., 2017, Done et al., 2018, Huang et al., 2019). The idea of voxelization is similar to rasterization except the point cloud is voxelized into 3D cubes rather than 2D pixels. Upon the voxelization result, supervoxelization can be generated to achieve an initial segmentation where the point cloud is divided into a number of relatively small clusters that can help extract more complex attributes in the following merging stage to obtain the final segmentation. Even though MLS data shares many similarities with TLS data, when segmenting MLS data, most methods aim at segmenting each individual object where ground filtering can help separate the objects that are connected via the ground. Then, additional processing would focus on recognizing and classifying each object (Che et al., 2019).

Recently, the research team published multiple peer-review papers focusing on point cloud segmentation (e.g., Mahmoudabadi et al., 2016, Che and Olsen, 2017, Che and Olsen, 2018, Che and Olsen, 2019, Che and Olsen (under review)). All these papers include a comprehensive literature review discussing the state of the art in point cloud segmentation. The most recent segmentation method that the research team developed is called *Vo-Norvana* including three primary steps: data organization, normal variation analysis (Norvana), and point clustering. In the first step, the input point cloud is partitioned into 3D tiles with overlaps between each other. Then it implements a new voxelization approach that mitigate the loss of information during voxelization process. Once the data is structured and organized, the point cloud is classified into smooth, rough, and invalid surfaces based on the local normal gradient via Norvana. Finally, for each class, a custom region growing is designed to cluster the point cloud into segments (Figure 2-19). Comparing against other existing segmentation approaches, *Vo-Norvana* is very efficient and can process hundreds of thousands of points per second. Additionally, it can cope with different types of surfaces and is not limited to pre-defined geometric primitives. Moreover, the versatility of *Vo-Norvana* is demonstrated via tests on TLS, MLS, ALS, UAS-SfM data. Last but not the least, unlike many other existing methods only being tested on a few small datasets, *Vo-Norvana* has been tested on large datasets including hundreds of millions of points at a city scale.

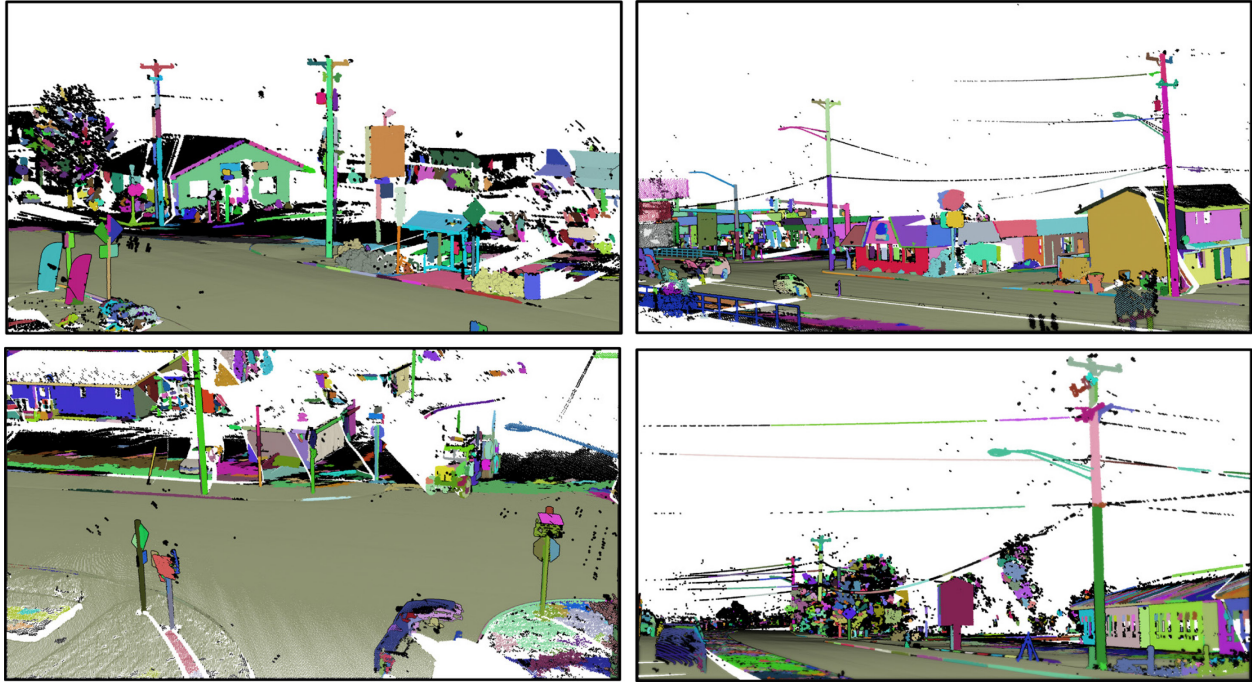


Figure 2-19: Example of *Vo-Norvana* segmentation on MLS dataset collected by ODOT using Leica Pegasus: Two where each color represents a segment while the unclassified points are colored in black

2.5.2.2 Road Marking Extraction

Road marking segmentation often starts by extracting the road surface to reduce the area to analyze. Subsequently, non-learning-based or binary classification are used to separate road markings from the pavement. Table 2-3 summarizes some of the most frequent approaches found in the literature. Similar to the road surface extraction, road marking segmentation can be divided into point-based approaches and image-based approaches.

Broadly speaking, the point-based road marking segmentations are categorized into (1) rasterization, (2) scanline, and (3) 3D-point based methods. Because the road surface can be assumed to be more or less planar over a localized area, many point-based approaches in the literature use the rasterization methods. 3D-point based methods can extract more detailed information, but they are often more computationally expensive. Because road markings are made of highly reflective materials that provide high intensity returns, most studies utilize the intensity information to extract road markings. Binary segmentation based on intensity is often used for the road marking segmentation (Guan et al. 2014; Guo et al. 2015; Yang et al. 2018, Guan et al. 2015, Riveiro et al. 2015; Soilán et al. 2017; Jung et al. 2019). The segmented road markings, however, may include noise with high intensity values. To suppress noise while preserving road markings, morphological operations (Guo et al. 2015; Guan et al. 2014; Kumar et al. 2014; Jung et al. 2019), median filters (Cheng et al. 2016; Riveiro et al. 2015), and high-pass filters (Cheng et al. 2016) are proposed in the literature. The Hough transform is commonly used to extract the linear-shaped road markings from the noise (Yang et al. 2012; Riveiro et al. 2015).

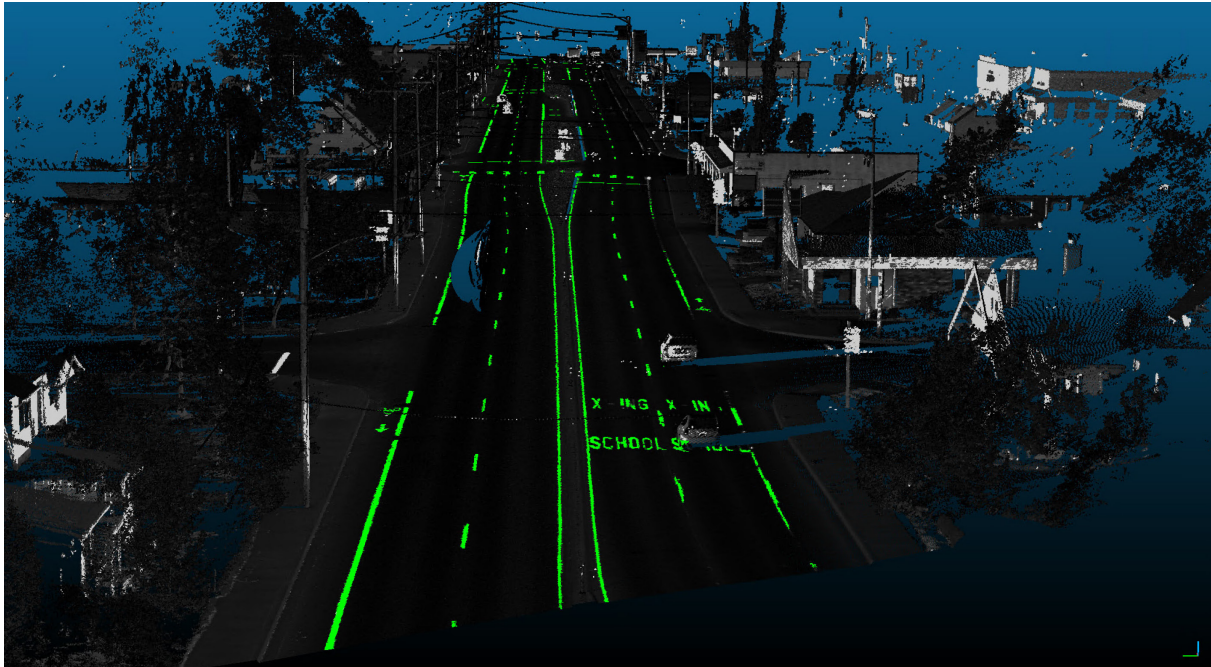
Jung et al. (2019) propose a statistical approach (i.e., the dip statistics) to filter out noise with unimodal distribution of intensity values. For 3D-point based analysis, Yu et al. (2014) propose a density-based filtering to filter out the points in neighborhoods of lower density as noise. For scanline analysis, Yan et al. (2016) employ a dynamic window median filter to minimize noise while preserving road marking edges. After noise filtering, road markings can be isolated to produce separate markings using a connected-component labeling process (Guo et al. 2015) or a region growing technique (Guan et al. 2015; Gheng et al. 2016).

Table 2-3: Summary of road marking segmentation methods

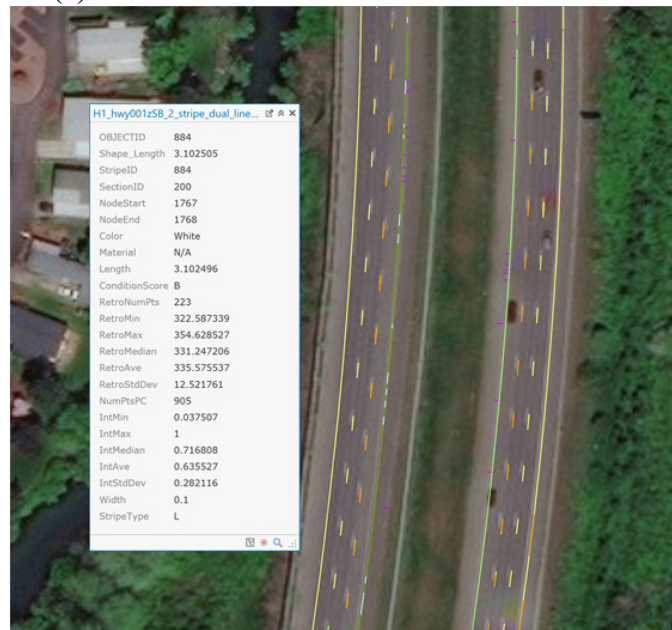
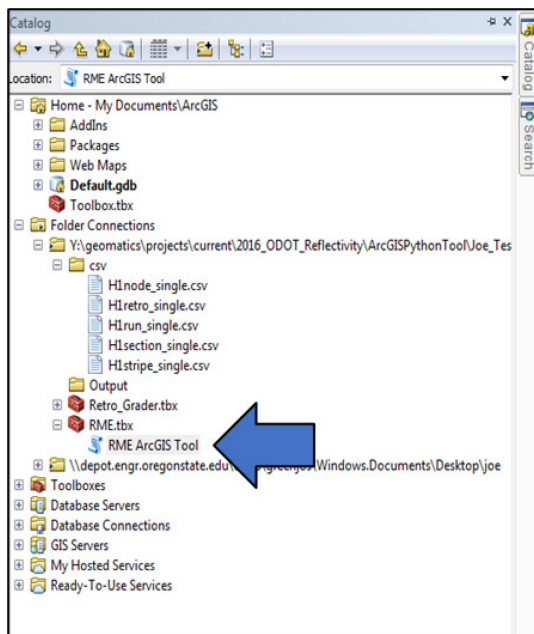
Study	Methods	Methods	Methods	Methods	Characteristics	Characteristics
	Raster	3D-based	Scanline	Image-based	Spectral Values	Geometry
Guan et al. 2014, Kumar et al. 2014	✓	-	-		✓	-
Guan et al. 2015 , Reveiro et al. 2015 , Jung et al. 2019 , Yang et al. 2012	✓	-	-		✓	✓
Guo et al. 2015, Cheng et al. 2017, Soilán et al. 2017, Wen et al. 2019	✓	-	-		✓	✓

Study	Methods	Methods	Methods	Methods	Characteristics	Characteristics
	Raster	3D-based	Scanline	Image-based	Spectral Values	Geometry
Yang et al. 2017	-	✓	-		✓	✓
Yu et al. 2014	-	✓	-		✓	✓
Yan et al. 2016	-	-	✓		✓	✓
Yang et al. 2018	-	-	✓		✓	✓
Azimi et al. 2019, Kurz et al. 2019, Yu, et al. 2021				✓	✓	✓

For SPR799 and continued research with PacTrans, the research team developed a road marking extraction tool (RoME) to segment complex road markings from mobile laser scanning data. Using the geometric information of the point cloud, the road surface is extracted by ground filtering, followed by slope filtering. Subsequently, the extracted road surface is rasterized into a 2D intensity image, and high-intensity pixels, which are most likely road markings, are segmented. Further, in order to reduce noise, the research team recently applied an unsharp masking technique, which outperformed the commonly used high-pass filtering technique. Figure 2-20a depicts the road markings segmented by the proposed approach using the point cloud acquired by the ODOT's Pegasus:Two system.



(a)



(b)

Figure 2-20: Example results from the RoME tool (ver. 2.1): (a) segmented road markings and (b) RoME Toolbox and its associated scripts within ArcCatalog.

RoME provides five output attribute tables (*Run*, *Section*, *Stripe*, *Node*, and *Retro*) in CSV (comma separated values) file format. These tables are linked through a set of IDs such that the attributes can be efficiently stored but still available throughout the various data layers. The generated CSV files can be imported into commercial GIS software or analyzed by themselves. The research team developed scripts using ArcGIS python

(arcpy) to automate this conversion and reconstruction of the geometry as a GIS layer. Upon export of the csv files from the RoME tool, users would then run the python scripts in sequence to create a file geodatabase containing the results of each run. While each run will likely be an individual section of highway, the final results can then be merged with previous runs for inclusion in a statewide TransGIS layer using the “merge” geoprocessing tool available within ArcGIS, as necessary. Figure 2-20b shows the RoME ArcGIS toolbox and the example of an output layer. ODOT GIS has subsequently expanded this toolset with additional tools such as classifying skip lines.

Image-based road marking segmentation using photogrammetric data is similar to the rasterization methods, but it is more challenging due to the lower resolution (usually larger than 0.2 m) that results in a loss of details as compared to the rasterized point cloud data (usually larger than 0.04 m). Early image-based road marking segmentation methods first extract road surface, then use non-learning-based algorithms, such as Otsu method (Otsu, 1979), for road marking segmentation (Jin et al. 2012). Azimi et al. 2018 proposed the first single-step learning-based method, called Aerial LaneNet approach, to segment the lane markings. For training, aerial images are cropped into smaller patches and fed into a network model. In the end, the segmented images are combined together to provide a complete image. They utilized a modified CNN to extract tiny lane-markings from aerial images. Further, they enhanced the accuracy of their approach by using wavelet transforms and a custom loss function to address the issue of unbalanced ratio between lane markings and non-lane markings.

2.5.3 Classification

2.5.3.1 General

Classification is the process of semantically labeling data to support further analysis of the scene. Broadly speaking, object classification can be categorized into rule-based or machine learning approaches.

Once the point cloud is segmented into objects, different objects can be identified by using simple rules based on the prior knowledge of the scene. For example, Fan et al. (2013) perform a supervoxel-based segmentation and use normals, height, color, intensity, and shape of an object as constraints for determining the classification. Given a more complex scene, Yang et al. (2015) segment the point clouds into supervoxels, which are merged based on their coordinates, geometric attributes, and color values. Subsequently, a set of rules are set for classification using geometric and contextual features based on prior knowledge for each object in the scene. Although a rule-based classification approach may require many parameters depending on the number of object types, these parameters are generally straightforward to choose. Further, the unsupervised model can be used without labeled data that is required to train the supervised models.

Supervised machine learning techniques have been widely used for object classification for both images and point cloud data. In general, different objects are first localized in the scene. Then, many geometric (e.g., normals, incidence angles) and radiometric (e.g., RGB, intensity) properties (called features) are extracted and fed into a machine to train a

model. Once the model is trained, it can be reused to predict the labels for new data. Commonly adopted machine learning techniques are Support Vector Machine (SVM), Random Forest (RF), and Neural Network (NN).

SVM is one of the most used machine learning algorithms that has been applied to many use cases. SVM uses an optimal separation line called a hyperplane to separate two or more different classes. The goal is to find the optimal hyperplane through training with carefully labeled data where the “answer” is given to the machine. For non-linear data, the algorithm uses kernel tricks to map the original feature space to some higher-dimensional space (Cortes and Vapnik, 1995). RF is a tree-based decision tool that uses root and branch-like models to identify possible classes. RF basically combines multiple individual decision trees that act as an ensemble. These multiple trees are trained together to solve the same problem. The output is the combination of mutually exclusive decisions, which outperforms individual decisions (Breiman, 2001).

NN or Artificial Neural Networks (ANN) is a prediction approach that mimics biological neurons in order to recognize relationships within data. It consists of artificial neurons (nodes) that are connected to form a network. These nodes are operators similar to a multi-linear regression models based on the properties (or features) fed into the algorithm. The connections between nodes are modeled using weights, where positive values result in an excitatory (stronger) connection, whereas negative weights lead to inhibitory (weaker) connections. The weights are obtained through a backpropagation process (Hecht-Nielsen, 1992) with multiple iterations of attempted prediction using the training data to identify which node connections (and subsequently) which combinations of features contribute most to obtain the “correct” answer based on the training data. The output of the training is then simply the sum of inputs modified by weights and can then be applied to similar data.

Deep learning has recently become a very popular topic across many disciplines due to its effectiveness in classification. Deep learning uses a similar mechanism to NN, but it forms much deeper layers, which have been achieved by overcoming overfitting and vanishing gradient issues that can occur when training networks. Convolutional Neural Networks (CNNs) are the primary architecture for classifying/labeling an image in deep learning approaches (Krizhevsky et al. 2012). Based on CNN, the Fully Convolutional Network (FCN) concept is proposed, enabling segmentation and classification of images containing multiple objects (e.g., Long et al. 2015).

To apply the CNN architectures for segmenting and classifying point cloud data, the point cloud data typically needs to be reorganized into a 2D image by rasterization. Another option to represent point cloud data in a 2D format is collecting images from the scene using a virtual camera, such that various geometric and radiometric information can be captured and fed to the deep learning model (e.g., Zhuang et al. 2015, Lawin et al. 2017, Qiu et al. 2018). Instead of organizing 3D data into 2D images, some other methods voxelize the point cloud data and use 3D CNN techniques that take voxels as input (e.g., Huang and You, 2016). Some advanced approaches voxelize the point cloud data at multiple scales. This enables the local features to be extracted and trained adaptively to different scales (e.g., Wang et al. 2018, Roynard et al. 2018).

PoinNet/PointNet++ (Qi et al. 2017 a,b), *CorDet* (Xu et al. 2021) and *PointCNN* (Li et al., 2018) are three approaches specific to use deep learning to classify a general scene in an unorganized point cloud. Recently, Wu et al. (2019) created an algorithm *PointConv*, which uses CNN to classify (semantic segmentation) an unorganized point cloud in 3D space by using a multi-layer perceptron (MLP) comparing neighboring points to a series of reference points to generate the necessary features, providing more flexibility. Note that while these approaches are very promising, most of these approaches have focused on basic objects in indoor scenes (e.g., chairs, tables, doors, ceiling, wall, floor, etc.) and use relatively small point cloud datasets (few million points), which poses difficulty in scaling them to typical mobile lidar point clouds (hundreds of millions of points).

Because of the substantial success of deep learning in semantical segmentation of the scene, there have been many approaches developed to use lidar data for autonomous driving and robotics (e.g., Zhou and Tuzel, 2018). When compared to MLS data, the point cloud data utilized in autonomous driving and robotics typically contain fewer and less accurate. Nevertheless, these techniques have potential for future use in the classification of MLS data (Che et al. 2019).

2.5.3.2 Road Marking Classification

In this step, segmented markings are further classified into different categories. Rule-based or machine learning approaches are commonly used for the classification. Template matching is one of the simplest approaches, which computes the sum of absolute difference (Guo et al. 2015) or cross-correlation (Yang et al. 2018) to measure the similarity between the marking and a specific template in a database. Cheng et al. (2016) propose a hierarchical decision tree to classify the various road markings, such as zebra crossing, dashed and continuous lines, arrows, diamonds, and words. Soilán et al. (2017) propose a hierarchical classification using the neural network to differentiate the arrow and rectangular markings. The correlation coefficient and Structural Similarity Index (SSIM) method are then used to classify five different types of arrows. Recently, deep-learning neural networks have shown promising results on the classification of road markings. Wen et al. 2019 propose a hybrid approach that uses the rule-based clustering for the large size road markings (lane markings and zebra crossings) and CNN classifier for the small, complex markings (dashed lines, texts, arrows, diamond, and triangle markings). More recently, many deep-learning approaches (Cheng et al. 2020; Ma et al. 2020; Wolf et al. 2021; Patel et al. 2021; Yu et al. 2022) adopt U-Net structure for a single-step learning based classification. The U-Net architecture consists of a contracting path and an expansive path. In the contracting path, the network extracts the high semantic, but lower resolution feature maps from input images through a series of down-sampling convolutions. In the expansive path, the network recovers the original resolution through a series of up-sampling convolutions of the feature maps. One important modification in U-Net is that it extrapolates the loss of border pixels during the decoding step by mirroring the corresponding input images in the encoding step (Ronneberger et al. 2015).

Unlike the point-based classification, most of the existing image-based approaches focus on the lane marking detection issues (Table 2-4), owing to the fact that the other types of

markings (symbols, letters, arrows) appear to be too small to detect their complex patterns from aerial images (Yu et al. 2021). Traditional machine learning approaches, such as Random Forest (Fischer et al. 2018) are still used to detect lane markings, but U-Net deep learning frameworks have recently received a lot of attention for image-based classification as well. Kurz et al. (2019) propose a modified CNN in which images are first chopped to the size of 512×512 pixels before being fed to the network as input. As a result, a binary pixel-wise mask is generated to distinguish between lane and non-lane marking pixels. In the final step, the chopped images are combined using a customized stitching algorithm. A list of road marking classification methods is summarized in Table 2-4. In the last column of Table 2-4, the features (e.g., bike lanes, crosswalk) relevant to this project are in bold.

Table 2-4: Summary of road marking classification methods

Study	Features	Features	Features	Features	Classification	Classified
	Raster	3D-based	Scan-line	Image-based		objects
Sakr, 2019	✓	-	-	-	3D CNN	Bike lane, Lane centerline, Road edge, Yellow single solid line, Yellow single broken line
Wolf et al. 2021	✓	-	-	-	U-net	Arrows, lane dividers, stop lines, crosswalk, bike lanes, barred areas
Cheng et al. 2020	✓	-	-	-	U-net	Lane markings

Study	Features	Features	Features	Features	Classification	Classified
	Raster	3D-based	Scan-line	Image-based		objects
Yao et al. 2021	✓	✓	-	-	GAF, PointNet++	Dashed lines, solid lines, arrows, protected lines
Patel et al. 2021	✓	-	-	-	U-net	Lane markings
Yu et al. 2015	-	✓	-	-	GAF, Deep Boltzmann Machine	Arrows, stop lines, centerlines, crosswalk , solid lines, pedestrain warning
Cheng et al. 2017	✓	-	-	-	Decision tree	Lane markings, Stoplines, dashed lines, zebra crosswalk , arrows, text, diamond

Study	Features	Features	Features	Features	Classification	Classified
	Raster	3D-based	Scan-line	Image-based		objects
Ma et al. 2020	✓	-	-	-	U-net, CapsuleNet	Crosswalk, text, solid lines, diamond, arrows, dashed lines
Fischer et al. 2018	-	-	-	✓	Random Forest	Lane markings
Azimi et al. 2019	-	-	-	✓	FCNN	Lane markings
Kurz et al. 2019	-	-	-	✓	Wavelet-enhanced CNN	Lane markings
Yu et al. 2022	-	-	-	✓	CapsuleNet	Lane markings

2.5.3.3 Other objects

The research team developed a framework to classify various urban objects. Prior to object classification, it is desirable to merge the segments at the object level as the segmentation often provides too many details. Initially, the input points are grouped to generate clusters whose maximum Euclidean distance to its nearest cluster is greater than the predefined distance. Seed segments with less than a predefined height are identified within each cluster. A region-growing based bottom-up segmentation algorithm was developed to identify individual objects in the same cluster. After which, a refinement step is performed to identify and merge the over-segmented objects into one segment. Subsequently, a total of 25 geometric features (e.g., linearity, planarity, scattering, etc.) are extracted first from the segments and used to train a Random Forest model (Machine Learning) to classify 6 different types of objects. Figure 2-21 shows an example of basic urban objects classified by the proposed framework.

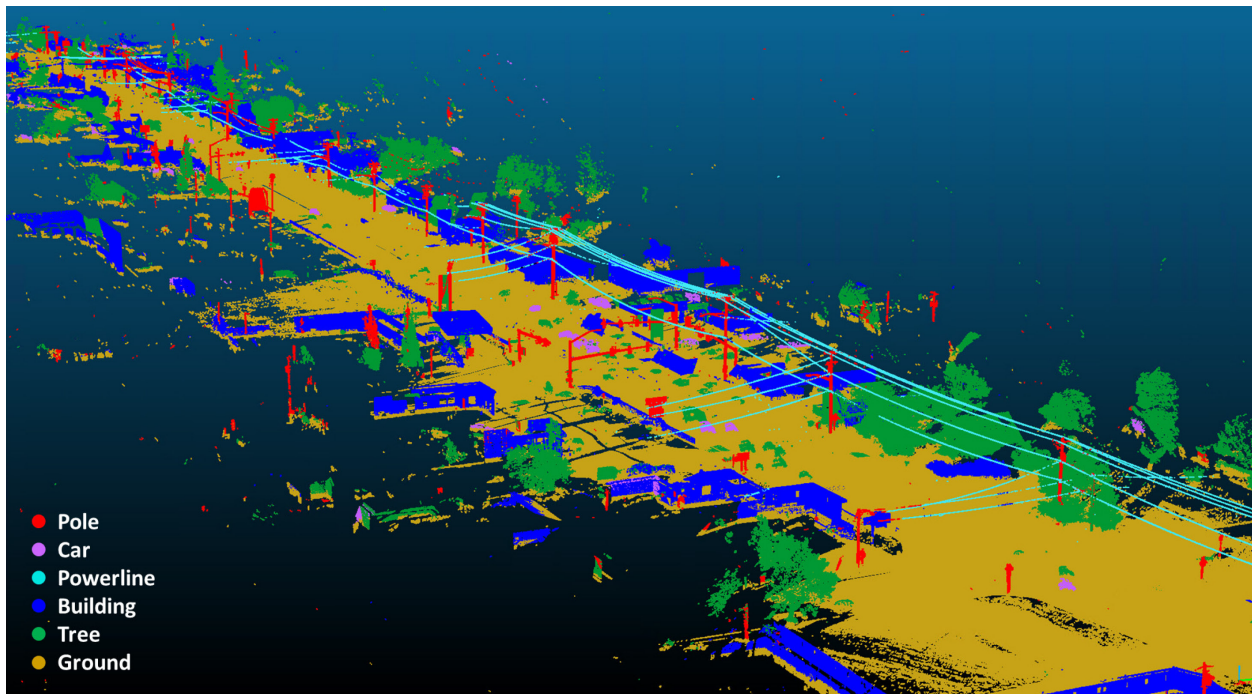


Figure 2-21: Classification of common urban objects within a point cloud collected by ODOT's Leica Pegasus:Two

Zaboli et al. (2019) explored the classification of mobile lidar data using several machine learning approaches, including K-nearest neighbors (KNN), Gaussian Naïve Bayes (GNB), Support Vector Machine, Multilayer Perceptron Neural Network, and Random Forest. Objects were classified as building, billboard, asphalt road, sidewalk, powerlines, tree, poles, and cars. Random Forest achieved the best results at 90% accuracy followed by KNN at 87%. SVM and MLP achieved similar accuracies close to 80%. GNB did not produce acceptable results (53% accuracy). Most algorithms (aside from GNB) performed best detecting asphalt roads, sidewalks and buildings and struggled the most detecting cars and trees.

2.6 ATTRIBUTION AND ANALYSIS

Other researchers have evaluated automated approaches to analyze point cloud data to extract information for safety analyses. These include:

- Road surface superelevation (Tsai and Ai, 2017)
- Horizontal Curve Identification and Measurement (Ai and Tsai, 2015)
- Cross slope measurement (Tsai et al. 2013; Shams et al. 2018 Shams et al. 2020)
- Sidewalk assessment for ADA compliance (Ai and Tsai 2016, Ai and Hou 2019, Ai and Hou 2020, Huo and Ai 2021)
- Sign Retro-reflectivity (Ai and Tsai 2016, Olsen et al. 2018a)
- Pavement marking retro-reflectivity (Che et al. 2019, Olsen et al. 2018b).
- Concrete joint faulting (Tsai et al. 2012).
- Pavement cracking, rutting, and debris detection (Hu et al. 2021, Ravi et al. 2021, Famili et al. 2021).

2.7 SUMMARY

The research team reviewed state-of-the-art point cloud-based and image-based methods for extracting transportation assets from point cloud and image-based datasets. The primary advantage of the point-based methods is its ability to detect details which enables classification of more diverse features. Specifically, for road markings, point-based methods mostly use intensity values for segmentation and classification, and thus are not sensitive to lighting conditions. Having reviewed the existing work for road boundary extraction, we found that most of the point-based methods rely on curb extraction. Unfortunately, curbs may not be available for some situations such as a rural area (Zhong et al. 2016) thereby limiting the use of these algorithms to only extracting well-defined, paved road surfaces in urban areas. Once road surfaces are detected, rasterization is found to be the most preferred method to segment the road markings because of their planarity (Table 2-3). Regarding classification, the U-Net-based deep learning framework has been widely adopted and modified over the last few years because it enables a single-step learning based end-to-end approach that integrates both segmentation and classification of objects such as road markings. However, one important limitation of most deep learning approaches is that they require a massive amount of training data, which often requires expensive and time-consuming labeling process that can sometimes be difficult to scale. Data augmentation approaches can aid in increasing the amount of training data by generating slightly modified copies of existing data.

The main advantage of aerial image-based approaches is that they can cover large areas in short time at a low cost. However, assets can sometimes be difficult to extract from aerial imagery given the resolution. For example, road markings in aerial images appear as tiny patterns; hence, many image-based methods can only extract simple lane markings. Further, aerial images often include shadows and occlusions due to nearby non-road objects, such as trees, bridges and buildings. These inconsistent radiometric conditions can lead to issues with the extraction, causing assets to be missed or incorrectly classified.

While many advances have occurred in the research and practitioner communities for feature extraction from lidar, several challenges still remain for feature extraction from point clouds or images.

- A significant portion of assets must still be extracted manually. Most automated research algorithms are not scalable and not yet ready for use in production and still require significant manual cleanup. Commercial software has provided tools to efficiently extract certain features (e.g., markings, curbs, roadway, signs) but generally require the user to click on a seed point for the object to be extracted to commence the operation. Hence, significant manual effort is still required. Users must have access to powerful computing resources and have training in specialized software to complete this effectively.
- Several approaches have been developed to extract road surface, curbs, and lane markings. However, there are limited approaches for extracting bike lanes, medians, and crosswalks. (While road marking extraction techniques have extracted cross walks, they have not explored classifying different types of crosswalks).
- AI (e.g., Machine Learning and Deep Learning) approaches have shown great promise in recent years. However, these techniques require substantial effort to train models. These models sometimes have difficulty scaling to datasets collected with different sensors or under different configurations. Most have only been tested/evaluated on relatively small, “idealized” datasets that include a very limited amount of features and do not necessarily represent real world conditions with substantial wear and tear.
- Many more tools exist for utilizing AI for image-based analysis compared with point cloud analysis. For example, AI has been used extensively for developing sign inventories from video logs of highways given that signs can be efficiently matched to templates based on shape and color. Many point cloud-based approaches rasterize the point cloud to utilize image-based tools, but suffer from the loss of detail in the process.
- Attribution of the extracted features is still largely a manual process and implemented separately in GIS.
- Given these challenges, many companies have adapted to a service-based approach where they run in-house code followed by manual processing to deliver a dataset rather than provide software to end-users to complete the extraction themselves.

Despite these challenges, remotes sensing technologies still provide powerful techniques to develop and maintain vast asset inventory databases and offer significant cost, quality, and safety benefits over current manual practices. The work in SPR850 significantly reduces the manual effort and improve the consistency of information extracted.

3.0 METHODOLOGY OVERVIEW

3.1 OVERVIEW

The research team developed an efficient workflow (Figure 3-1) and supporting tools (Table 3-1) to identify, extract and attribute bike lanes, pedestrian crosswalks, and medians from mobile lidar data. Early in the project, aerial photogrammetric datasets were considered; however, the resolution and image quality of statewide data sources (e.g., OSIP- Oregon Statewide Imagery Program, NAIP- National Agriculture Imagery Program) was insufficient (60 cm +). In many cases, given the blockage and lighting contrast created by the dense tree canopy present in much of Oregon, features were hard to identify in the orthophotographs. The overall methodology can be summarized into five general steps:

1. **Site Selection** includes identifying sites with appropriate data to support the training and testing of the methodology. Five proposed sites have been selected throughout the state to represent a variety of scenes (e.g., urban, suburban, rural, scenic). (Section 4.1). A sampling of data from these sites were used for training (Section 4.2) and Testing (Section 7).
2. **Data organization and indexing.** Data were organized for each site, including mobile lidar, photogrammetric (e.g., OSIP), and bike/ped GIS asset data (e.g., TransGIS). To facilitate rapid data querying, the research team developed a simple method to link lidar trajectory to highway and milepost for efficient data management and querying from the mobile lidar archives. (Section 5.1)
3. In the **data preprocessing** step (Section 5.2), the research team leveraged and expanded upon previous work including *Vo-norvana* segmentation, road surface extraction, and curb extraction, which helped focus the development in the subsequent steps on priority features of road markings, bike lanes, crosswalks, and medians.
4. Once the point cloud data are preprocessed, they are fed into the next step for **classification and extraction of features** (Section 6.0) to identify the locations of the features. Both parametric and machine learning approaches were utilized to improve the classification performance. Sections 5.0 and 6.0 more details of each step. The algorithms were evaluated with diverse datasets compiled in the *Site Selection* to ensure that they are reliable and function for the most common situations throughout Oregon (Section 7.0).
5. **GIS editing and updating.** Finally, the GIS layers can be annotated with relevant attributes for integration to TransGIS. The outputs can be utilized in several ways. One possibility is to use the output layers as basemaps to digitize features of interest. With the clearly defined curbs and road markings, many other features of the road can be identified much easier compared with drawing from a point cloud or aerial imagery. Alternatively, GIS scripts can be developed to support data cleanup and additional feature extraction.

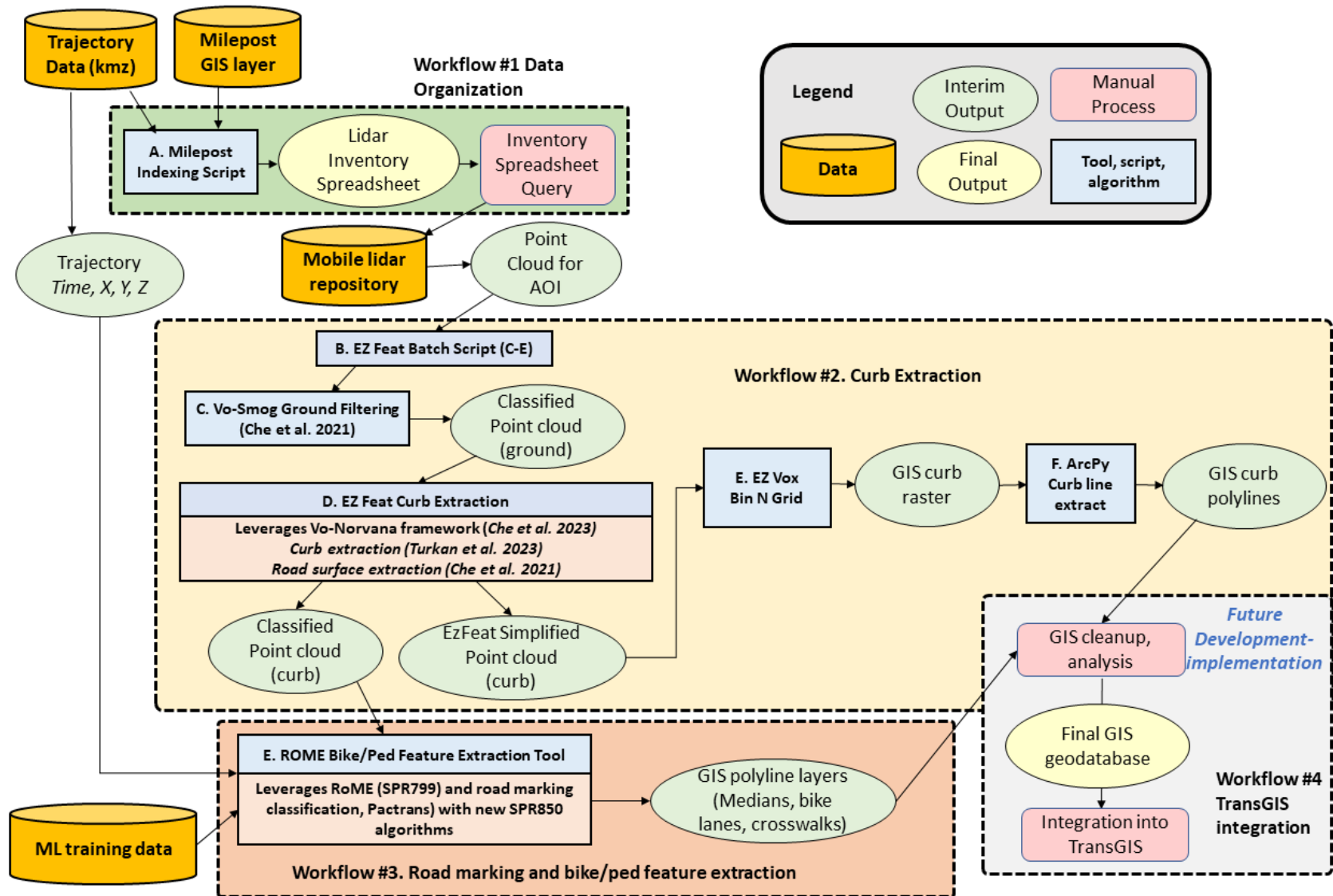


Figure 3-1. Overview of the tools, workflows, and methodologies utilized in SPR850.

3.2 TOOL DEVELOPMENT

The algorithms were packaged into a functional software tool that input ODOT's mobile lidar point cloud and trajectory data to produce a GIS geodatabase identifying the locations of bike lanes, crosswalks, and medians. These tool packages are identified as workflows in Figure 3-1. The output layers are suitable for ingestion into ODOT's TransGIS platform after some data cleanup and quality control. For evaluation of the developed tools, the extraction results were rigorously tested and optimized on a variety of mobile lidar datasets collected across the state (Section 7.0) as well as with testing by ODOT personnel.

The developed tools are research prototypes. Note that although the tools are functional and have been tested thoroughly to provide good results with typical datasets, limited budget was available to highly customize the tools and GUI or extensively test them to robustly handle a wide range of potential issues with the data or circumstances that arise. These tools are currently only licensed for internal use by Oregon DOT and cannot be shared externally due to licensing constraints.

The first tool package (Workflow #1) consists of an ArcPy script that links mobile lidar trajectory files with ODOT's milepost layer to identify which mobile lidar las/laz files correspond with the area of interest (AOI). The algorithm for this tool package is described in Section 5.1. A user manual is provided in Appendix A.

The second tool package (Workflow #2) consists of several executables developed by EZDataMD LLC. These tools are linked by a batch script for efficient processing of multiple files. Both a metric and international feet version of the script are provided. The executables include Vo-Smog Ground filtering, EZ Feat Curb Extraction, and Bin N Grid. Outputs from each executable are organized into different directories to facilitate efficient review and quality control. At the completion of these scripts, output rasters with multiple bands in bsq (binary sequential) format are available for import into GIS. Lastly, an arcpy script converts the raster layers into simplified vector features. Appendix B provides the user manual for these tools.

The third tool package (Workflow #3) is the Road Object and Marking Extraction (ROME) tool. This tool has a simple GUI (Figure 3-2) that can be used by people without significant experience in point cloud processing. Several extraction tools and procedures are packaged into a single platform, combining all key developments described in Section 6 for simple, but efficient processing. The tool inputs point clouds and exports annotated GIS shapefiles with attributes of classification, length, and width. Appendix C provides the user manual for this tool.

These layers can then subsequently be cleaned up in GIS through both manual operations and scripts. However, this manual effort is substantially reduced compared with digitization from scratch. linked to milepost information through GIS spatial operators, such that it is easier for ODOT to integrate the research product to their existing approaches of organizing geospatial data. Section 8.0 describes the implementation of this research in more detail.

User manuals are provided in Appendices A-C to improve the usability of the software tools. Lastly, a training session was conducted at the end of the project where ODOT personnel could practice using the tool with sample data, learn more about how the algorithm works, and provide feedback on its capabilities and limitations.

Table 3-1. Tools provided with SPR850 and associated inputs and outputs.

ID	Tool	Description	Inputs	Outputs
A	Milepost Indexing python script	Tool to help identify which mobile lidar files correspond with a specific location.	Kmz, and las/laz files	Inventory database as xml.
B	EZFeat batch script	Batch script to execute steps C-E. It will automatically pass input/output files between executables.	See below	See below
C	Vo-Smog Ground filtering exe	Tool to classify points as ground or non-ground. (Che et al. 2021).	las/laz files	Ground classified las\laz file.
D	EZFeat curb extraction exe	Tool which classifies curb points in the mobile lidar dataset.	Ground classified las/laz files	Curb classified las\laz file. Simplified las\laz file with curbs only and gaps filled in.
E	EZ Vox bin n grid exe	Tool to create raster layers from point clouds.	Simplified curb las laz	.bsq raster
F	Curb line extractor ArcPy script	ArcGIS python script to convert rasters to linework.	.bil raster	.shp
G	Rome 3.7.2 GUI	Tool to extract road markings, perform retro-reflectivity analysis, and extract crosswalks, bike lanes, and medians. Initially developed in SPR799, significantly updated through Pactrans and SPR850.	las/laz files (ideally curb-classified for optimal results)	.shp Provides attributes of retroreflectivity, classification, length, and width.
H	ArcPy scripts	Tools to help with editing and analysis of pavement marking GIS files. Delivered in SPR799 with additional tools developed by ODOT GIS.	.gdb	.gdb

Optional Tools

The following tool can be used to perform ground filtering before running the other tools.

Rambo Ground Filter	Software delivered to ODOT for SPR809 for rockfall analysis. Can be used as a ground filter in place of Vo-Smog.	las/laz files	Classified las\laz file (Classification 2)
----------------------------	--	---------------	--

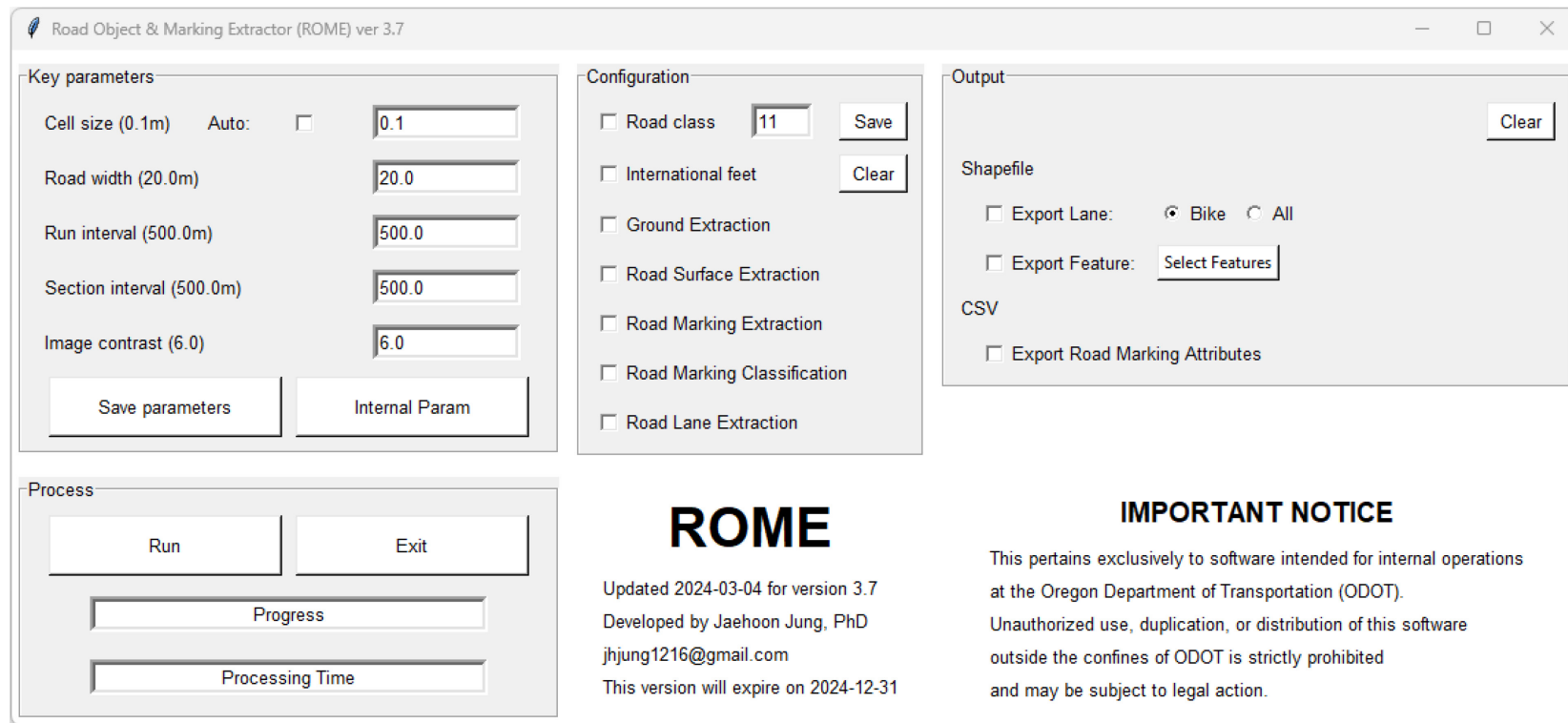


Figure 3-2: ROME v3.7 GUI developed by the research team. See Appendix C for more details.

4.0 TRAINING DATASET

4.1 SITE SELECTION

In conjunction with the TAC, the research team identified five sites for training and testing the classification algorithms to be developed for bike/ped safety analysis (Table 4-1). An additional site (Newport) was identified as a backup. These sites represent a variety of scenes (e.g., urban, suburban, rural, scenic) where the presence, style, and complexity of these assets vary. All of the proposed sites include multiple bicycle facilities, medians, and marked crosswalks including those with and without connecting ADA ramps. Notably, most sites contain connections with local roads in addition to the ODOT highway(s). To the extent possible, features visible on those segments were also extracted to serve as a pilot for integrating databases between agencies.

For each site, the most recent mobile lidar data available through Geometronics and bicycle and pedestrian asset data through TransGIS were utilized. The data were carefully reviewed against street view and aerial imagery to verify data accuracy and reliability. The data were also checked to ensure that they do not contain features that were planned to be built as well as to ensure the features are consistent with what was present at the site of the time of the mobile lidar or photogrammetric data collection.

Table 4-1: Proposed sites to support the algorithm training and testing.

Site ID	Location	Setting	Highways	ODOT Internal Highway #	Approximate Length (Miles)	Social Equity Rating (ODOT, 2022)
1	Downtown Salem (Figure 4-1)	Urban	• HWY 221 (MP16–21), • HWY 99EB (MP0–8.5)	• 150 • 072	13.5	Medium-high / high disparity
2	Tigard and King City (Figure 4-2)	Busy Suburban	• HWY 99W (MP8–13) • HWY 141 (MP2–7)	• 091 • 141	10.0	Medium-high / high disparity
3	Hillsborough (Figure 4-3)	Less dense suburban with some near rural	• HWY 029 (MP 0–18) • HWY 102 (MP 88–91)	• 029 • 102	21.0	High disparity
4	Albany (Figure 4-4)	Moderate size city	• HWY 99E (MP0–8)	• 058	8.0	Medium-high / high disparity
5	Portland (Figure 4-5)	Small, rural city	• HWY 213 (82nd Ave)	• 068	5.3	Varies substantially. Low to high disparity
Alt.	Newport (Figure 4-6)	Small Coastal Community	• HWY 101 (MP 134–140)	• 009	6.0	Medium-high / high disparity

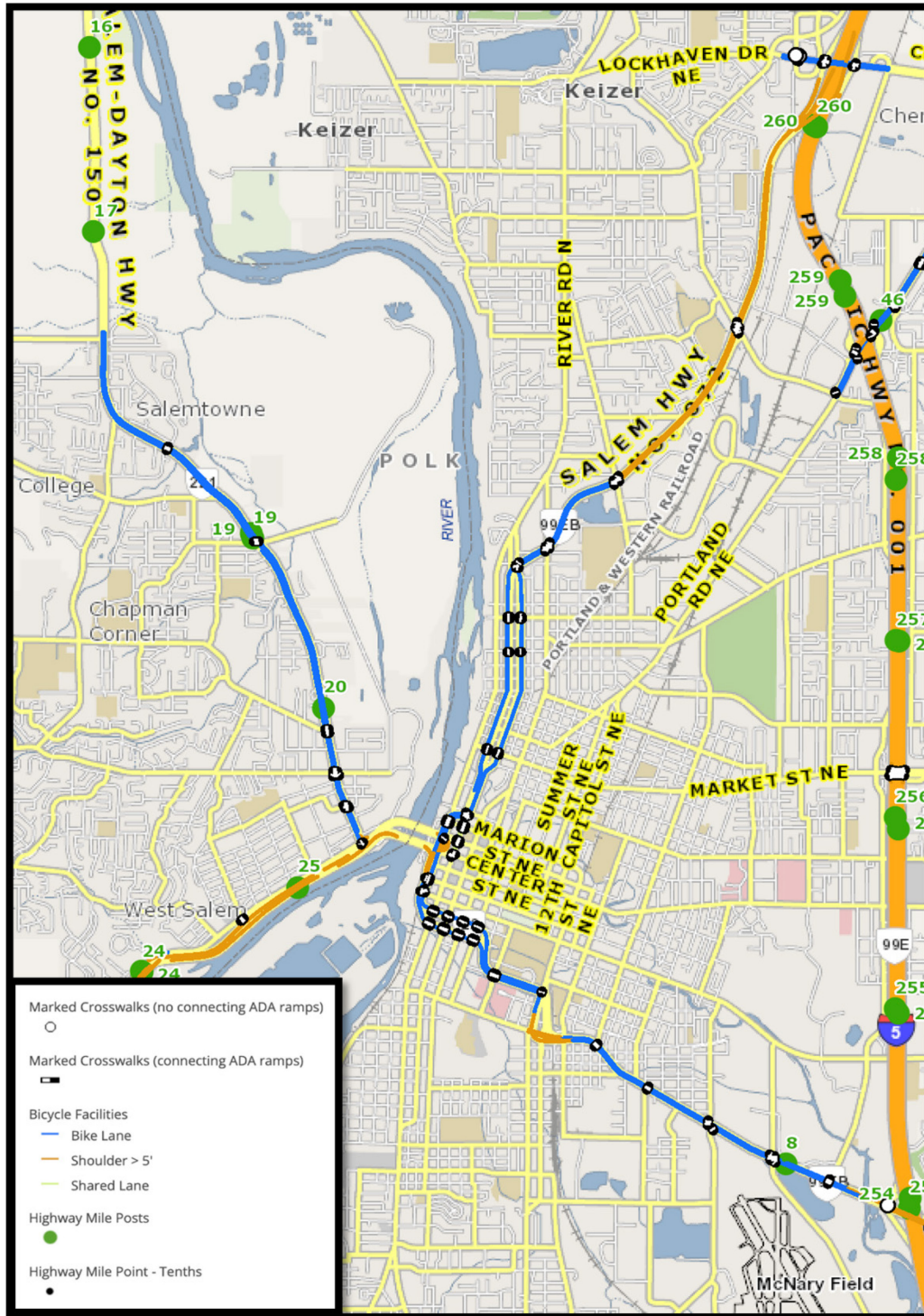


Figure 4-1: Data Collection Site #1 in Downtown Salem, Oregon consisting of an urban area between mileposts 16 – 21 on Highway 221 as well as a section of Highway 99 EB between mileposts 0.0 to 8.5.

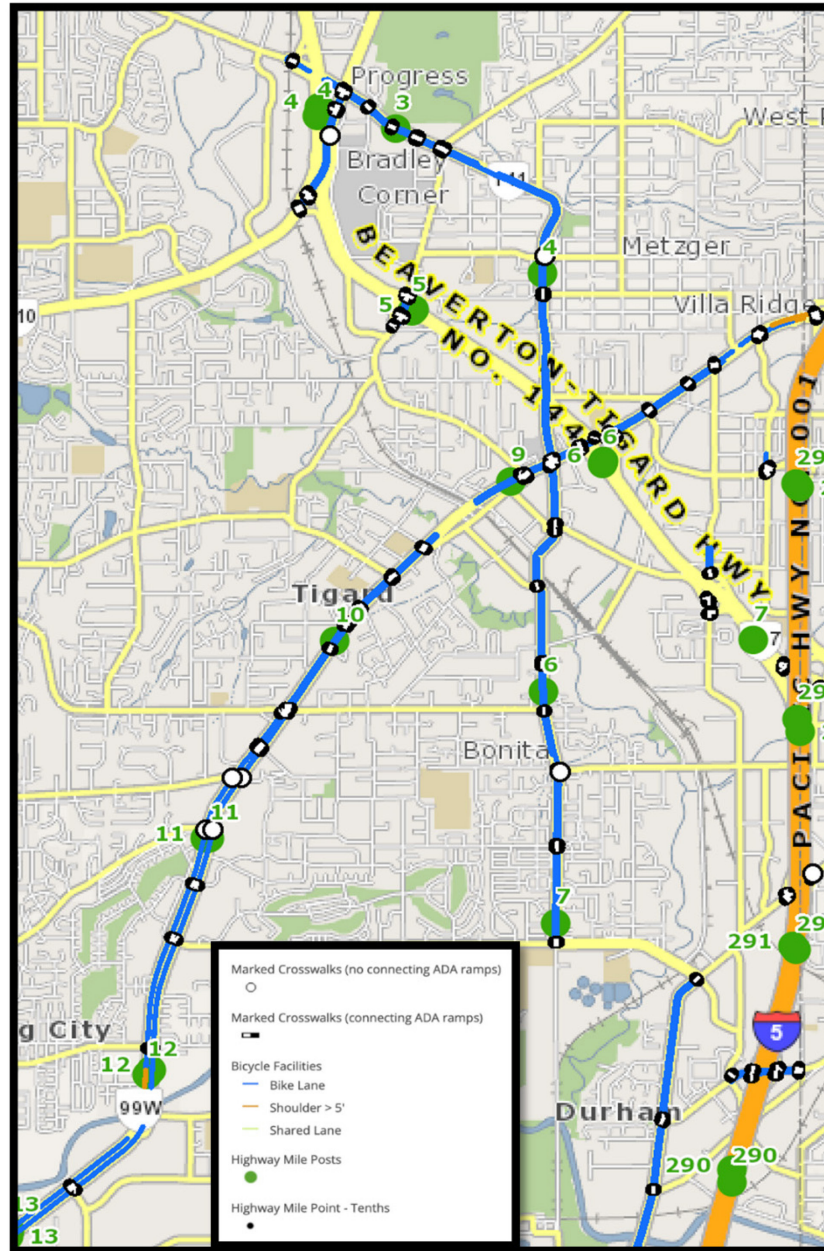


Figure 4-2: Data Collection Site #1 in Tigard Oregon consisting of mileposts 8 to 13 on Highway 99W, and from mileposts 2 to 7 on Highway 141. This site includes communities in Tigard and King City.

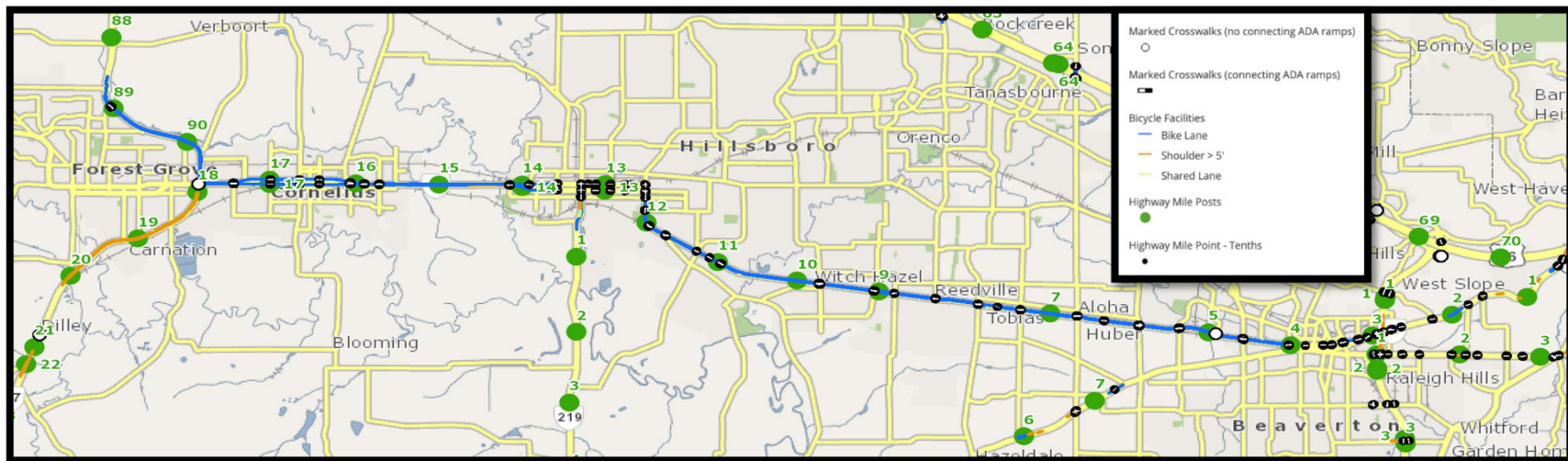


Figure 4-3: Data collection site #3 between mileposts 0 and 18 on Tualatin Highway No. 029, and includes a section of Nehalem Highway No. 102, between milepost 88 to the intersection with Tualatin Highway No. 029. This site includes communities in Forest Grove, Hillsboro, and Beaverton.



Figure 4-4: Data collection site #4. Site#4 consisting of mileposts 2 to 8 on Highway 99E, and the sections of Santiam Highway 16 and Albany – Junction City Highway between I5 and 99E.

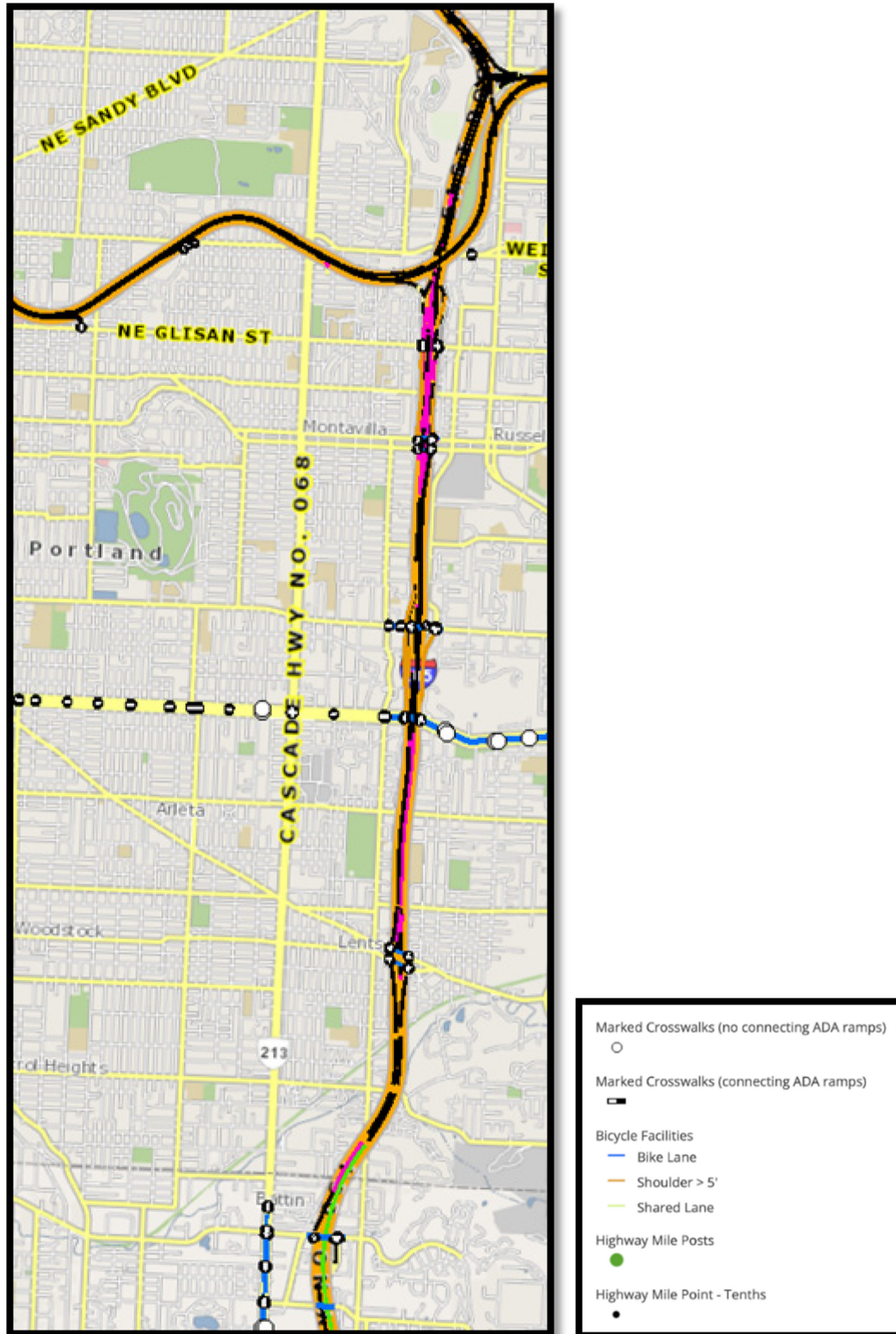


Figure 4-5: Data collection sites #5 on Highway 213 (82nd Ave) in Portland, OR.



Figure 4-6: The alternate site is located on Oregon Coast Highway 101 between mileposts 134 and 140.

4.2 DATA ANNOTATION

4.2.1 Categories

Features of interest include crosswalks, bike lanes, and medians. After a review of ODOT design manuals, subcategories were identified for each (Figure 4-7). Then, exploring the wide range of mobile lidar datasets tested in this research, the most common types were identified and prioritized for the feature extraction.

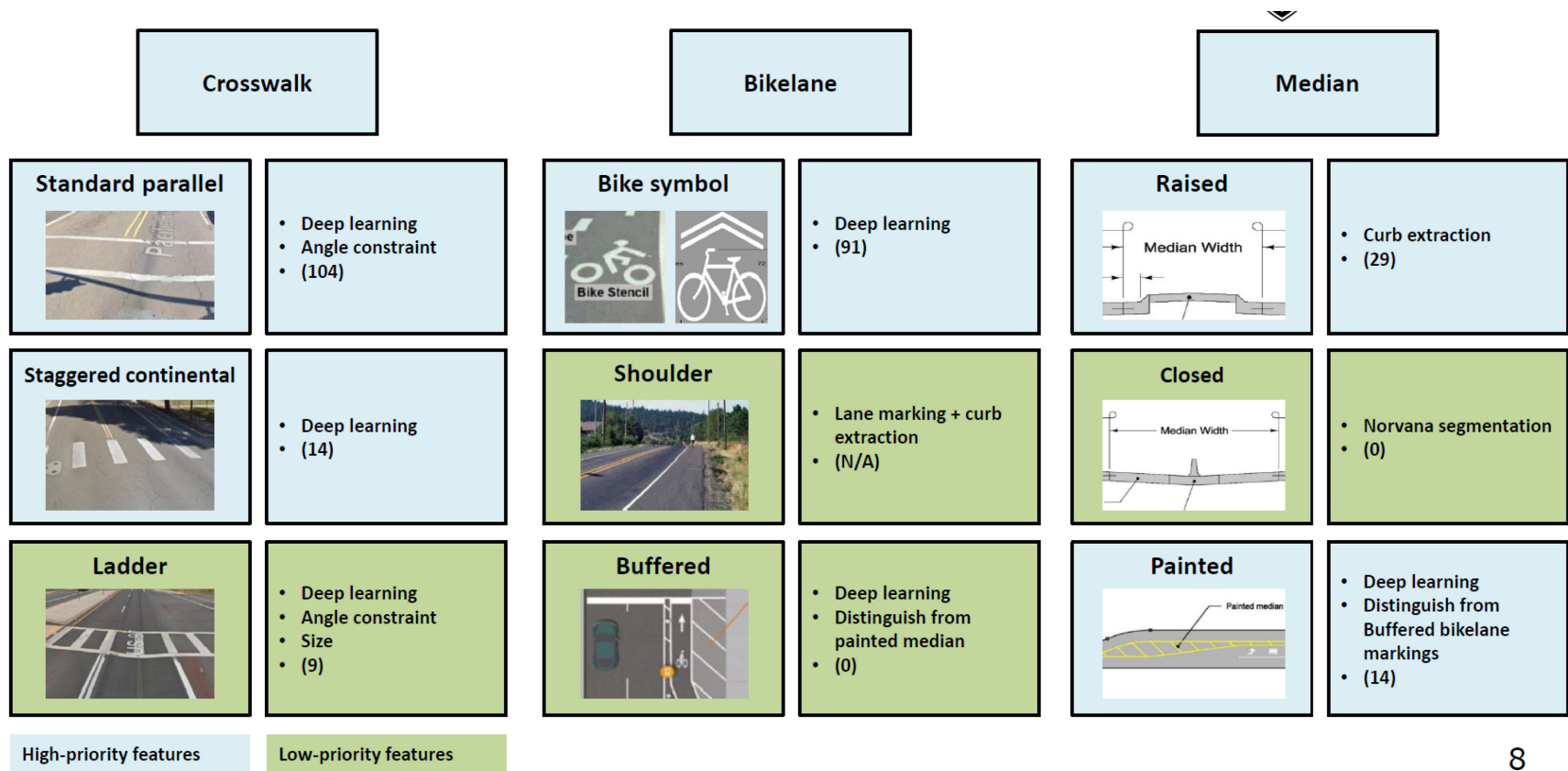
- *Bike lane* markings can be separated from other types of lane markings by detecting the bike symbols. The detected bike lane markings will be used to segment the portion of the road surface corresponding to bike lanes from the lidar data, and the relevant attributes, such as lane width, will be extracted.
- *Crosswalks* can be divided into standard parallel, staggered continental, and ladder types. They have different geometry for different purposes and should be classified further to extract relevant attributes, such as lane width and cluster information.
- *Medians* can be divided into closed (barrier), raised, and painted. Because the feature detection will be performed based on the 2D semantic segmentation, further work is necessary to extract 3D information for closed and raised medians, such as height.

4.2.2 Attribution

Table 4-2 shows the level of details for the features explored through this project. The basic classification (level 0) of this table follows the standard ASPRS LAS classification. The key features, bikeway, median, crosswalk, curb, that are specific to this project are highlighted in different colors. Note that not all features and attributes could not be reliably extracted from the mobile lidar data. As the research progressed, the research team determined which features and attributes are feasible to extract from the mobile lidar data for integration into TransGIS. Hence, the current version of the tools described in Section 3.0 do not extract all features in the table. However, the training dataset can be utilized to support further research in developing algorithms to extract those features.

4.2.3 Data Files

For labeling, the research team initially investigated 170 las files, containing 42 roadway features (Table 4-2). Prior to labeling, each dataset was explored in detail to identify how frequently each object appeared (Table 4-3). Appendix E contains the detailed list of features in the dataset. Based on this analysis, the list of files was then refined to 33 las files for the annotation process. Table 4-4 summarizes the number of labeled features in each file incorporated into the training dataset.



8

Figure 4-7: Summary of pedestrian features common on ODOT highways. Lower priority features are either less common or more difficult to extract.

Table 4-2: Feature types and relevant attributes

Features				Potential Attributes
Level 0 (ASPRS LAS 1.4R15 2019)	Level 1	Level 2	Level 3	
Ground	Bikeway	Bike lane with stencil		Lane width, Confidence score
		Shoulder		Lane width, Confidence score
		Shared lane w/ sharrow		Lane width, Confidence score
	Road markings	Lane marking	Line	Line width, transverse vs longitudinal, color, intensity statistics
			Protected line	
			Bike line	Line width, Confidence score
			Protected bike line	Confidence score
		Dash line	Dash line	Line width
			Bike dash line	Line width, Confidence score
		Crosswalk	Standard parallel	Line width, Confidence score
			Staggered continental	Line width, # clusters, Confidence score
			Ladder	Confidence score
		Arrow	Left	Width, length, intensity statistics
			Straight	Width, length, intensity statistics
			Right	Width, length, intensity statistics
		Letter		
		Others	Bike stencil	Width, length
			Bike sharrow	Width, length
			Diamond	Width, length
	Road surface			Longitudinal and cross slope, Roughness
	Curb			Length, height
	Medians	Depressed		Height, width, Confidence score
		Closed (barrier)	Concrete	Height, width, Confidence score
			Steel	Height, width, Confidence score
			Plastic	Height, width, Confidence score
		Raised		Height, Width, Confidence score
		Painted		Width, Confidence score

Table 4-3: Occurrences of features in the explored datasets showing the counts of available features in all files as well as only the labeled files.

Category	Symbol	Label	Label Number	Ext_Class	Count (All)	Count (Labeled)
Crosswalk	Parallel White Lines	Standard Parallel Crosswalk	20	110130	104	24
Crosswalk	Staggered Continental	Staggered Continental Crosswalk	21	110131	14	9
Crosswalk	White Ladder	Ladder Crosswalk	22	110132	9	7
Median	Raised	Raised Median	30	110140	28	12
Median	Closed	Closed Median	31	110141	0	0
Median	Painted	Painted Median	32	110142	14	4
Bike Lane	Bike	Bike Symbol	40	110150	91	28
Bike Lane	Double White Stripes	Standard Bike Lane	41	110151	14	5
Bike Lane	Painted Area around Bike Lane	Buffered Bike Lane	42	110152	0	0

Table 4-4: Occurrences of features in the labeled datasets

Labeled Data ID	File ID	Crosswalk			Median			Bike lane			Lane Marking	
		Std. Parallel	Stag. Cont.	Ladder	Raised	Closed	Painted	Bike Symbol	Std. Bike Lane	Buff. Bike Lane	White Solid	White Dash
SiteID_1_Hwy99EB												
4	1_HWY99EB_07	1						1			1	-
5	1_HWY99EB_13	1						1			1	-
6	1_HWY99EB_19			1				1			1	-
7	1_HWY99EB_20	1						1			1	-
8	1_HWY99EB_24	1						1			1	-
SiteID_1_Hwy221												
13	1_HWY221_06	1		1	1			1			1	
24	1_HWY221_08	1		1	1			1			1	1
25	1_HWY221_10	1		1	1			1			1	
26	1_HWY221_12			1	1			1			1	1
27	1_HWY221_17	1			1			1			1	
SiteID_2_Hwy99W												
23	2_HW99WSB_34	1	1		?			1			1	1
28	2_HW99WNB_18	1	1		1			1			1	1
29	2_HW99WNB_21							1			1	1
30	2_HW99WNB_36				1		1	1			1	
SiteID_2_Hwy141												
20	2_HW141_14	1	1				1	1			1	1
21	2_HW141_20	1						1	1		1	1
22	2_HW141_22	1						1			1	1
SiteID_3_Hwy29												
14	3_HW29_35	1						1	1		1	1
15	3_HW29_59	1			1			1			1	1
SiteID_3_Hwy102												

Labeled Data ID	File ID	Crosswalk			Median			Bike lane			Lane Marking	
		Std. Parallel	Stag. Cont.	Ladder	Raised	Closed	Painted	Bike Symbol	Std. Bike Lane	Buff. Bike Lane	White Solid	White Dash
9	3_HW102_0	1	1					1			1	1
10	3_HW102_1							1			1	
11	3_HW102_4	1						1			1	1
12	3_HW102_5		1		1							
SiteID_4_Hwy99E												
31	4_HW99E_44	1						1	1		1	1
32	4_HW99E_46	1						1	1		1	1
33	4_HW99E_10		1					1			1	
SiteID_5_Hwy213												
1	5_HW213SB_18	1					1					
2	5_HW213SB_26	1	1					1				1
3	5_HW213SB_30		1		1							
16	5_HW213NB_8	1						1	1			1
17	5_HW213NB_24	1			1		1	1				
18	5_HW213NB_26		1	1								
19	5_HW213NB_28	1		1	1						1	

4.2.4 Procedure

Maptek PointStudio software was utilized to label the markings into the aforementioned categories for 33 individual laz files (Figure 4-8 and Figure 4-9). First, these datasets were processed in the RoME v2.0 software in order to provide an efficient starting point for the manual delineation. Next, a Lasso tool was used to meticulously select points belonging to a marking or feature to correct the RoME output. Care was taken to only select the marking. Then the appropriate classification from Table 4-1 was assigned to the selected points. This process was repeated to extract all relevant features in the dataset. The results were then reviewed by other members of the project team. If necessary, the point annotations were adjusted to correct any issues. For example, the bulk of the annotations were initially performed in a 2D plan view. This resulted in errors with misclassifying signs located within medians as medians (e.g., Figure 4-10). These errors were corrected for the final database during the review process.

4.2.5 Utilization in machine learning

These training datasets were utilized to train a deep learning model for semantic segmentation of markings. Note that in order to ensure high classification performance, deep learning requires a massive amount of manually-labeled data for training the model. The deep learning classification is also sensitive to class imbalance. In the majority of roadway scenes acquired by a mobile lidar system, the road surface and lane markings occupy a very high proportion of the data compared with bike symbols, pedestrian crosswalks, and medians. If the labeled data is biased towards a certain class, the deep learning will also be biased towards the same class during the training. The trained model would assume any input data belongs to the majority class, which would result in inaccurate classification for the minority classes. To improve the class imbalance, a variety of methods, such as data filtering, data augmentation, balanced weighting, and a modified loss function that focus on minority classes, were incorporated into the training procedure.

Several deep learning approaches were explored during the training process and algorithm development including U-Net, Alex-Net, and modified VGG6 CNN approach. Ultimately, the modified VGG6 CNN approach was selected for those portions of the extraction process utilizing machine learning as it showed the best performance. Section 6.0 describes which portions of the process rely on deep learning and how that is integrated with geometric rules.

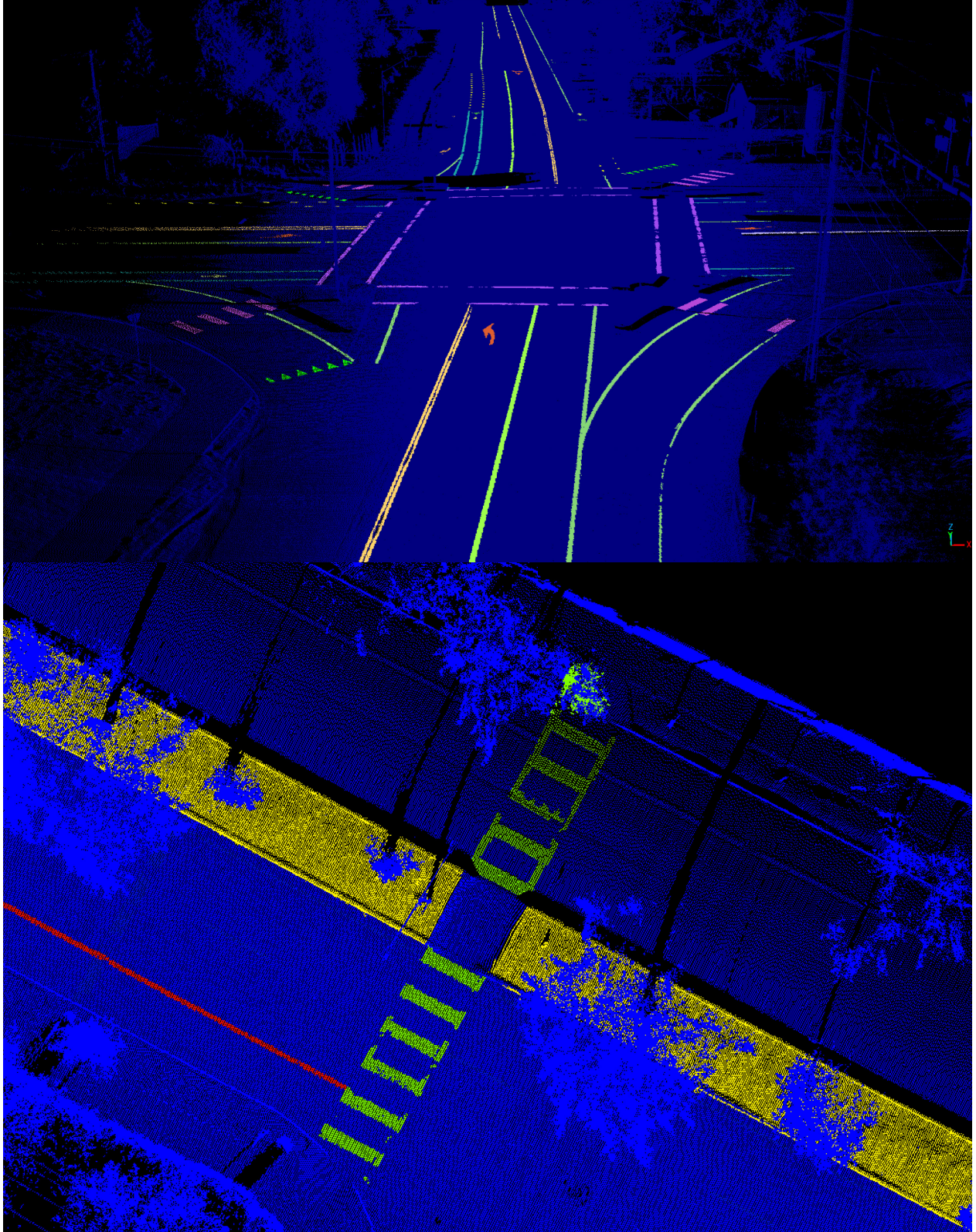


Figure 4-8: Example annotated point cloud with each class of marking identified with a unique color.

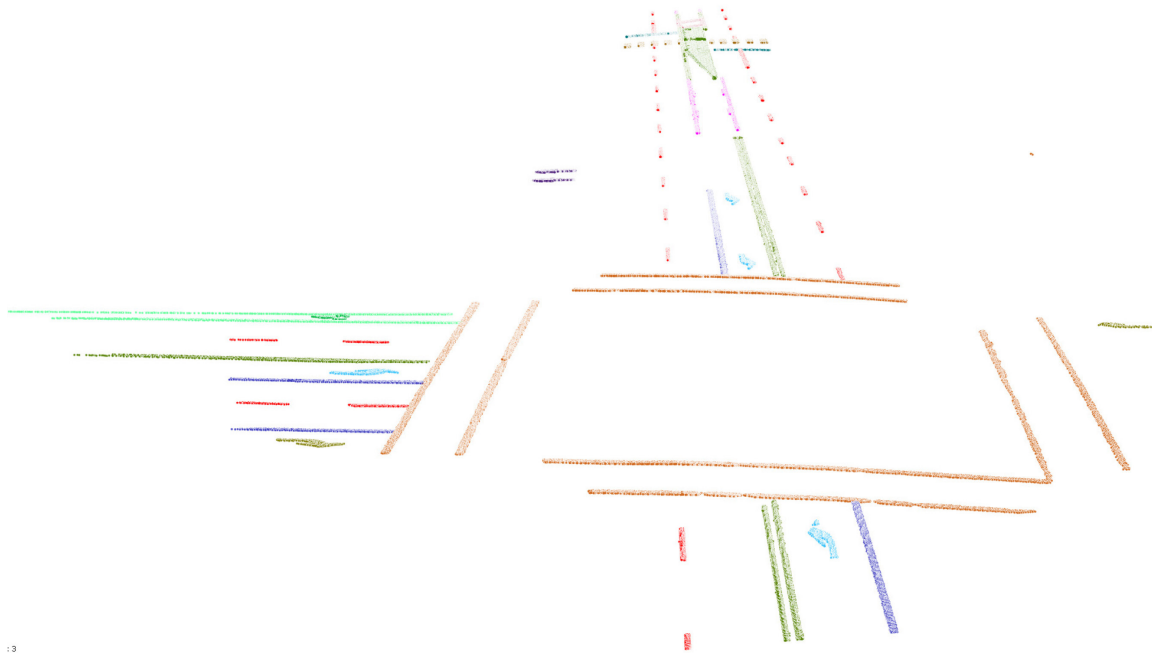


Figure 4-9: Example annotated point cloud with each class of marking identified with a unique color. (Annotated features only).

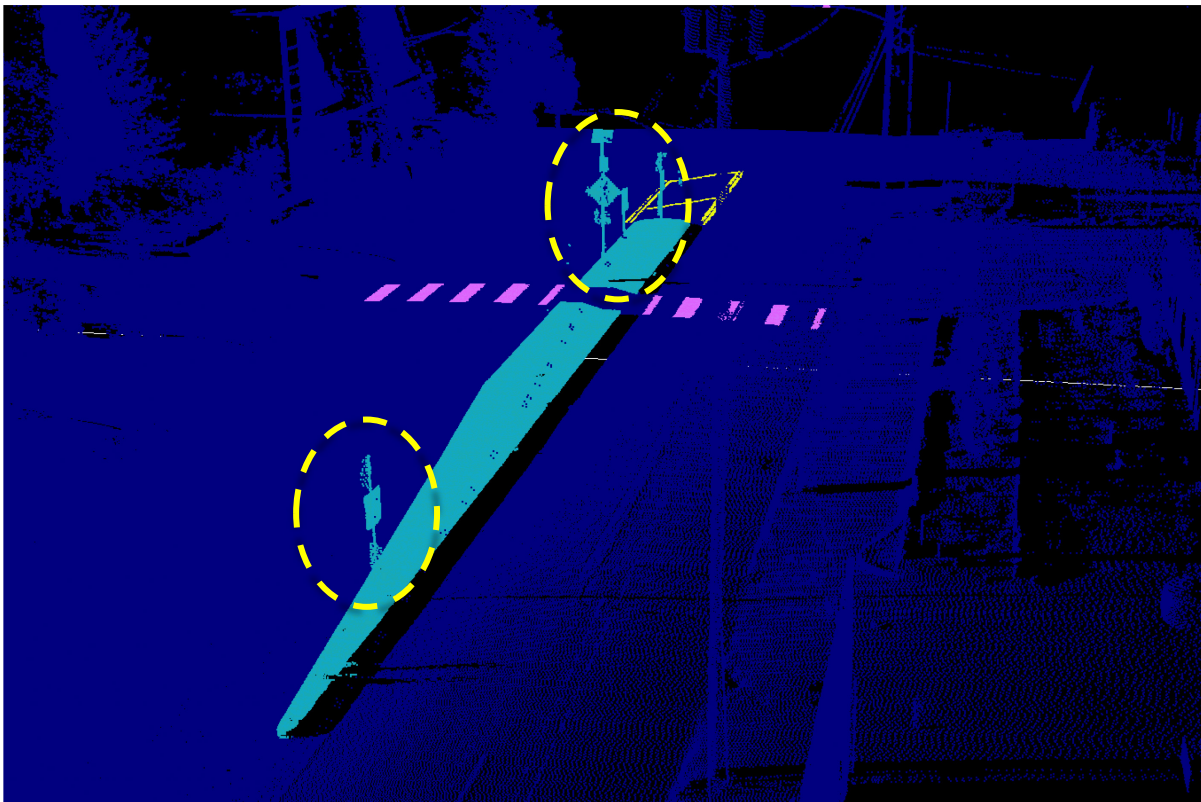


Figure 4-10: Example errors in initial plan view annotation that were corrected.

5.0 DATA PREPARATION

5.1 DATA ORGANIZATION AND INDEXING

The research team developed an automated trajectory-matching approach, linking the mobile lidar trajectory to ODOT's GIS highway milepost layer. The following steps are followed in the workflow (Figure 5-1):

1. The algorithm inputs the milepost geodatabase (e.g., tenths or hundredths of a mile) as well as MLS trajectory in KML format.
2. Both input data need to be projected to the same coordinate system.
3. Then the milepost data is converted to polylines based on the *Linear Reference Method Key* and the *milepoint* attributes. Because a KML trajectory file can contain multiple MLS passes, the MLS trajectory is split into individual polyline features such that each pass can be processed separately to improve subsequent querying processes.
4. The milepost coordinates are relatively approximate (based on GIS features digitized from satellite imagery); whereas the MLS trajectory is of high accuracy directly georeferenced with GNSS, IMU, and other sensors, and, in some cases, refined by applying ground control. As a result, there can be significant (tens of meters) spatial offset between the two datasets (Figure 5-2). Moreover, the MLS data corresponding to the trajectory can cover multiple lanes captured on the same highway. As a result, each individual polyline feature from the MLS trajectory needs to be aligned to the polylines generated from the milepost data (Figure 5-3). In this process, the search distance (the distance used to search for match candidate features) is set as 30 m (98 ft) by default to match the lines.
5. The extracted information from the milepost data is assigned to the end points of the polyline extracted from the trajectory data (representing the start and end points of an MLS pass). In this case, the program developed by the research team took the closest milepost point within the given search radius of 100 m (328 ft) to obtain this information.
6. This approach was implemented into a python script/toolbox (running in ArcGIS environment) that ODOT Geometrics and TransGIS can run on mobile lidar data that has been acquired to improve the management of the vast catalogues of mobile lidar data.

Once the milepost geodatabase and the trajectory files are fully registered, a GIS feature class can be created from a subsampled version of the trajectory to serve as an index/look up table to identify the location of mobile lidar datasets corresponding to a given highway and milepost range. The information can be organized and exported to an excel spreadsheet/table for data management applications (e.g., query, metadata, etc.) where filtering/querying operations can be executed. Further, the matching results can also be potentially mapped (with interpolation as necessary) as attributes for objects extracted in ArcGIS with similar procedures.

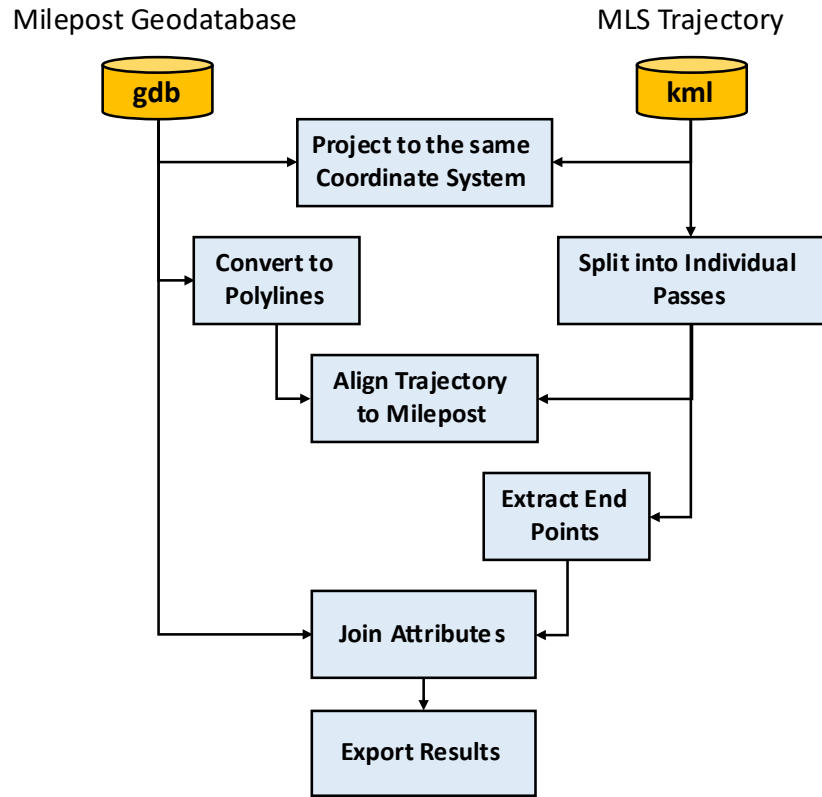


Figure 5-1: Workflow for lidar milepost indexing.

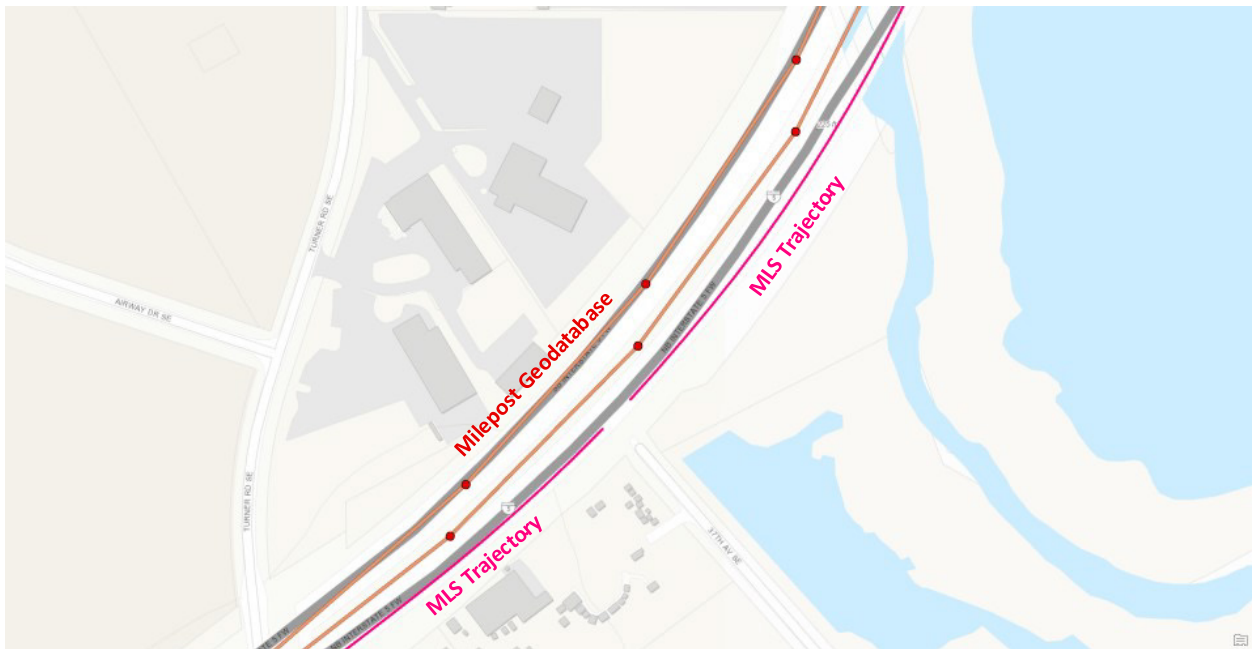


Figure 5-2: Visualization of the misalignment between the MLS trajectory and polylines generated from the lower accuracy milepost geodatabase.

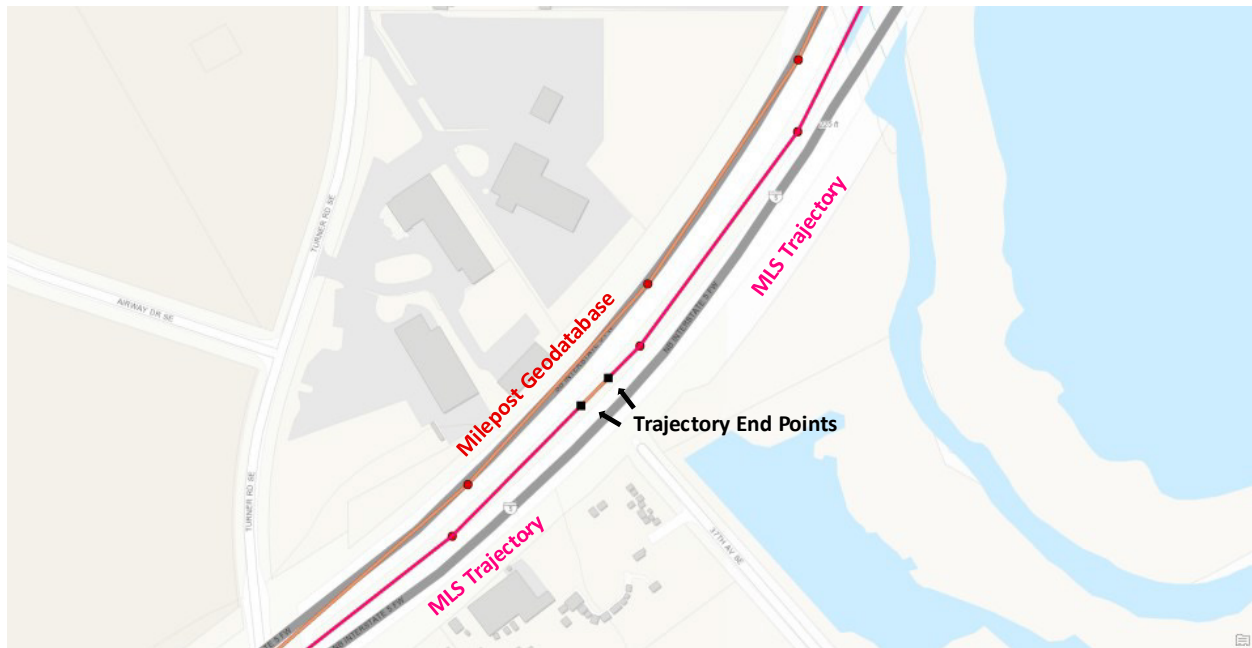


Figure 5-3: Alignment between the MLS trajectory and the milepost geodatabase.

5.2 DATA PREPROCESSING

Data preprocessing is necessary to achieve the goal of feature extraction and characterization. Such processes not only reduce the data volume to significantly increase the computational performance, but also simplify the data with additional attributes and semantic information extracted from the point cloud. In this work, the data preprocessing includes: profile density estimation (Section 5.2.1), ground filtering and curb extraction (Section 5.2.2), and road surface extraction (Section 5.2.3).

5.2.1 Profile density estimation

Because the scanner used in the Leica Pegasus:Two system is a 2D profiler which collects data with a constant angular increment (~ 0.07 degree), the point density decreases as the range and angle of incidence increase. Although the dual-profiler configuration is able to mitigate the low point density to some extent, given the fact that the second profiler collects data from a similar viewpoint, the point cloud data can still suffer from low point density for objects at range far from the scanner. While collecting data with multiple passes on different lanes can improve the point density, simply combining the data from multiple passes before further data processing can substantially increase the data volume and lower the computational performance. In addition, the geo-referencing errors between passes can result in lower precision in a local area. Moreover, for extraction methods for some features relying on intensity such as pavement markings, a rigorous radiometric calibration or normalization needs to take place before merging the point clouds because the intensity is a function of range, angle of incidence, and material. Therefore, the research team provided a tool that can initially process the mobile lidar data on each individual

pass independently to maximize the efficiency and minimize the errors and issues that can occur during the data merging process.

To record the quality or confidence of the feature extraction, a profile density estimation approach was developed and used to tag the estimated point density for each individual point (Figure 5-4). The point density can be divided into two components: along the profile and along the trajectory. The point spacing along the trajectory is a function of the driving speed, which is usually consistent in a data collection job and can be calculated from the trajectory file directly. On the other hand, because the time stamp and data source (if multiple profilers or passes are involved) and of each point from the mobile lidar point cloud are recorded in the data stored in LAS format according to the ASPRS LAS specification, the research team first sorts the point cloud based on time stamps. Then to estimate the point density along the profile at a data point, its nearest neighbor in terms of recording time from the same data source on both directions are searched, and the shorter distance is taken as the profile density. The profile density is recorded in the point cloud file as an LAS “extra bytes” field throughout the point cloud processing. The density can serve as a QA/QC or confidence metric.

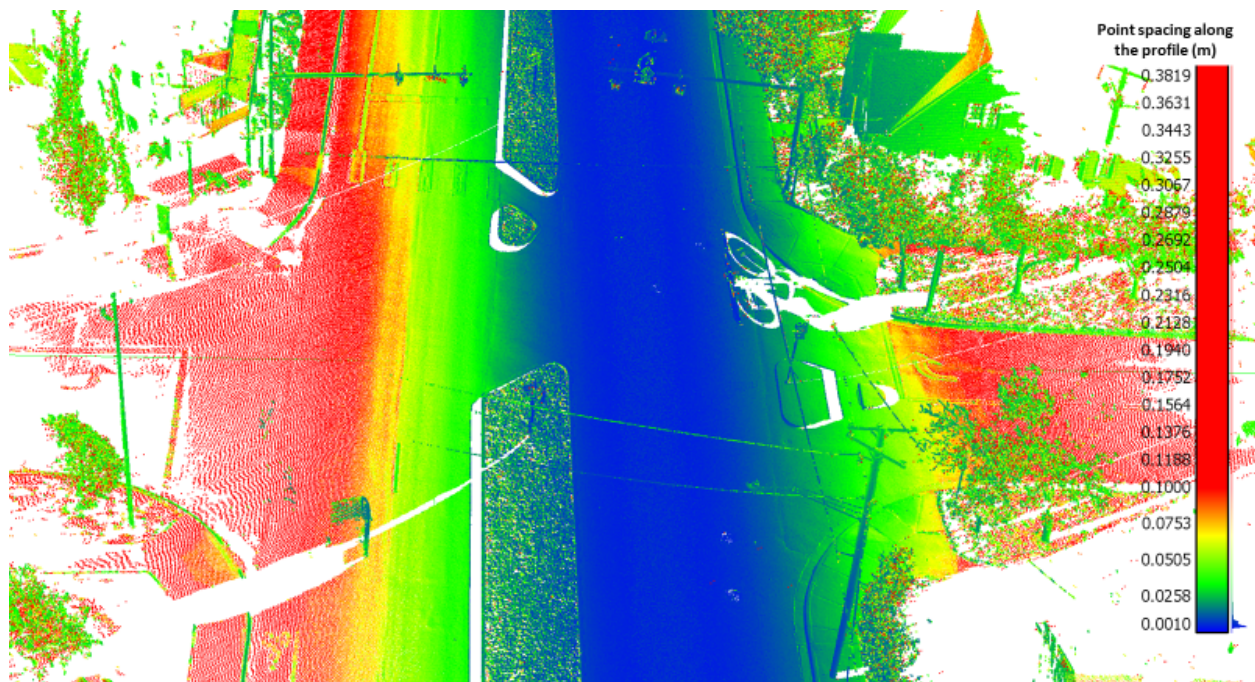


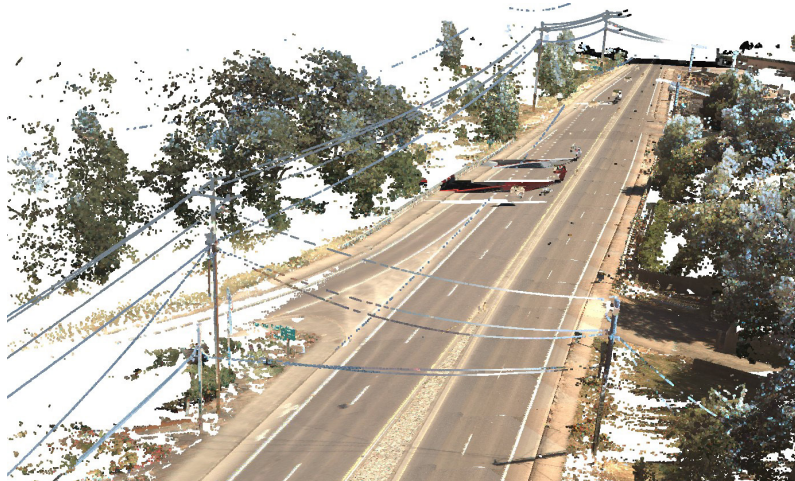
Figure 5-4: Example of profile density estimation for mobile lidar data.

5.2.2 Ground filtering

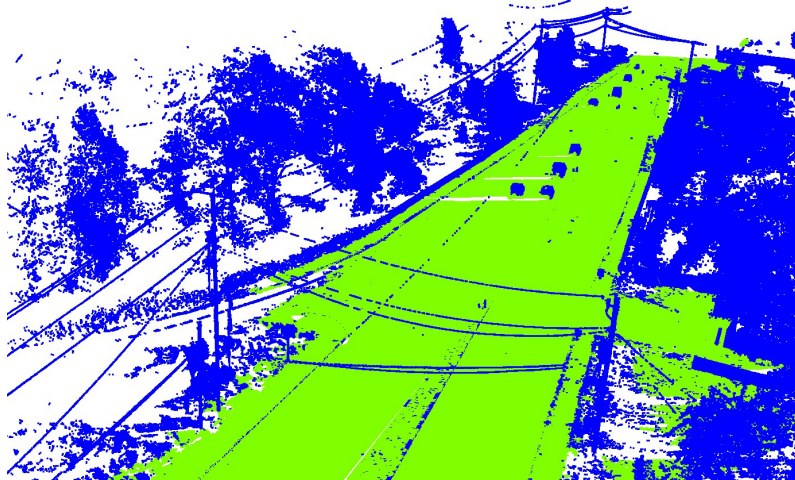
The curb detection procedure is not limited to any specific ground filter in terms of the strategies and definition of the ground surface. In this case, the highly efficient *Vo-SmoG* ground filtering (Che et al., 2021) approach developed by the team is adopted in the framework. This versatile ground filtering technique can be applied to many types of lidar or photogrammetric point cloud datasets. Ultimately, it classifies points as ground (2) and non-ground (1). Ground filtering is discussed in further detail in Section 2.5.1.1.

5.2.3 Curb extraction

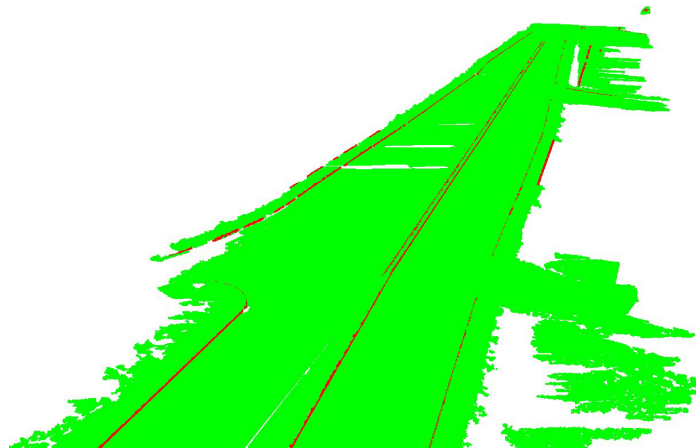
A curb detection algorithm developed previously by the research team is incorporated into the framework. The curb detection inputs a point cloud that is classified as ground and non-ground (Figure 5-5b). Because ground filters tend to remove steep surfaces such as the curb face as non-ground objects that technically are part of the ground, the first step of the curb detection is to add potential curb points back to the ground points. Then, the elevation change on the ground surface is computed locally and compared against a user-specified range of elevation change for the curb (0.02 – 0.3 m or 0.8 – 11.8 in.). With such a range, the curb detection can cope with some positioning errors and noise, as well as a variety of types of curbs often present within a scene (e.g., typical curb stones or the low curb delineators in the middle of the road in Figure 5-5a). Then the slope of each curb candidate point is computed to remove the flat areas, and the remainder of the curb candidates are clustered into groups. After the small clusters are removed, the average linearity of each curb cluster is estimated assuming the curb is close to linear in a local area. The curb detection result is finally mapped to the original point cloud with a classification ID (Figure 5-5c) while keeping the other classification information (e.g., ground and non-ground) for other feature extraction and analysis. For example, with the curb points serving as road boundaries, the road surface can be classified from the ground points, which is a necessary step in road marking extraction. Figure 5-6 shows an example of a simplified point cloud generated with curb points only.



(a)



(b)



(c)

Figure 5-5: Example curb detection results: (a) MLS data in RGB color; (b) Ground filtering result from *Vo-SmoG* where green points represent ground points; and (c) curb detection result (red points)

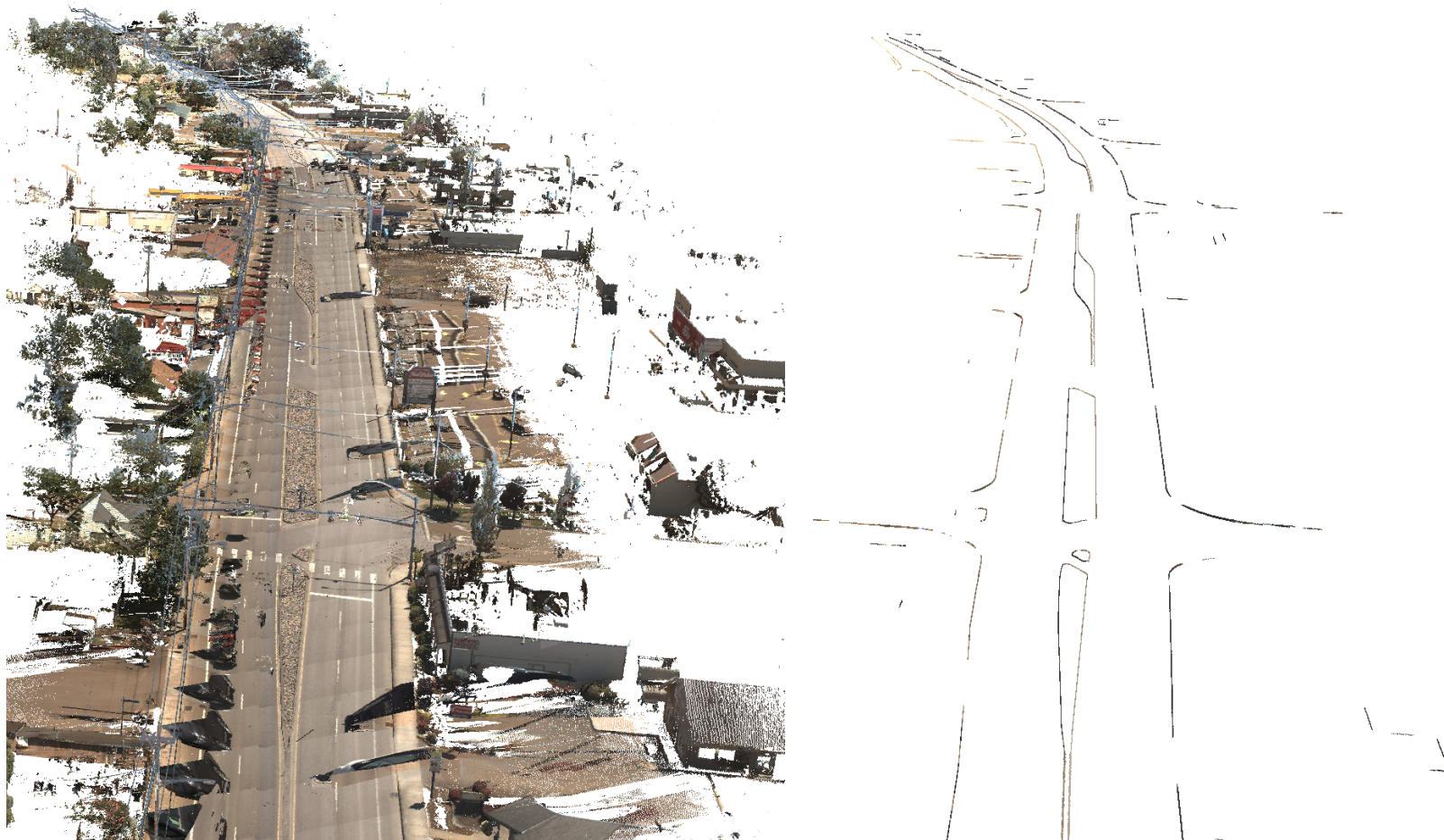


Figure 5-6. Example curb extraction. Left: Full dataset, Right Extracted curb points only.

5.2.4 Road Surface Extraction

The *Vo-Norvana* segmentation developed by the research team described in section 2.5.2 is able to detect smooth surfaces such as the road pavement. However, because of the existence of curb ramps, the native segmentation approach would connect the sidewalk and road surface and merges them into the same segment. In another prior work done by the research team (Che et al., 2021) called *VROOM*, the segmentation result was used to separate the ground and non-ground points and classify the road surface. By providing the width of a lane/car, the accessible area of a car can be detected because a typical curb ramp is usually narrower than a vehicle. At the time of the prior work, the team did not implement curb extraction in the workflow and the segmentation approach was based off of a specialized approach, *Mo-Norvana*, specifically developed for MLS data (Che & Olsen, 2019). However, *Mo-Norvana* heavily relies on the accuracy of the trajectory reconstruction, and it is sensitive to noise and rough surfaces. To improve the road surface extraction, the research team leveraged the curb extraction results including the *Vo-SmoG* ground filtering introduced in the previous section to help define the road boundary and provide the road candidates (ground points). This is especially important for suburb and urban areas as there are more moving objects making it challenging to identify the accessible areas without distinguishing those moving objects. In addition, the latest segmentation approach, *Vo-Norvana*, was also adopted in the road surface extraction to improve the robustness of the process. Once the road surface is extracted, the road marking extraction and classification processes can operate to the road surface layer only, which can improve the accuracy and efficiency. Figure 3.11 shows an example of a road surface segmented by the *VROOM*.

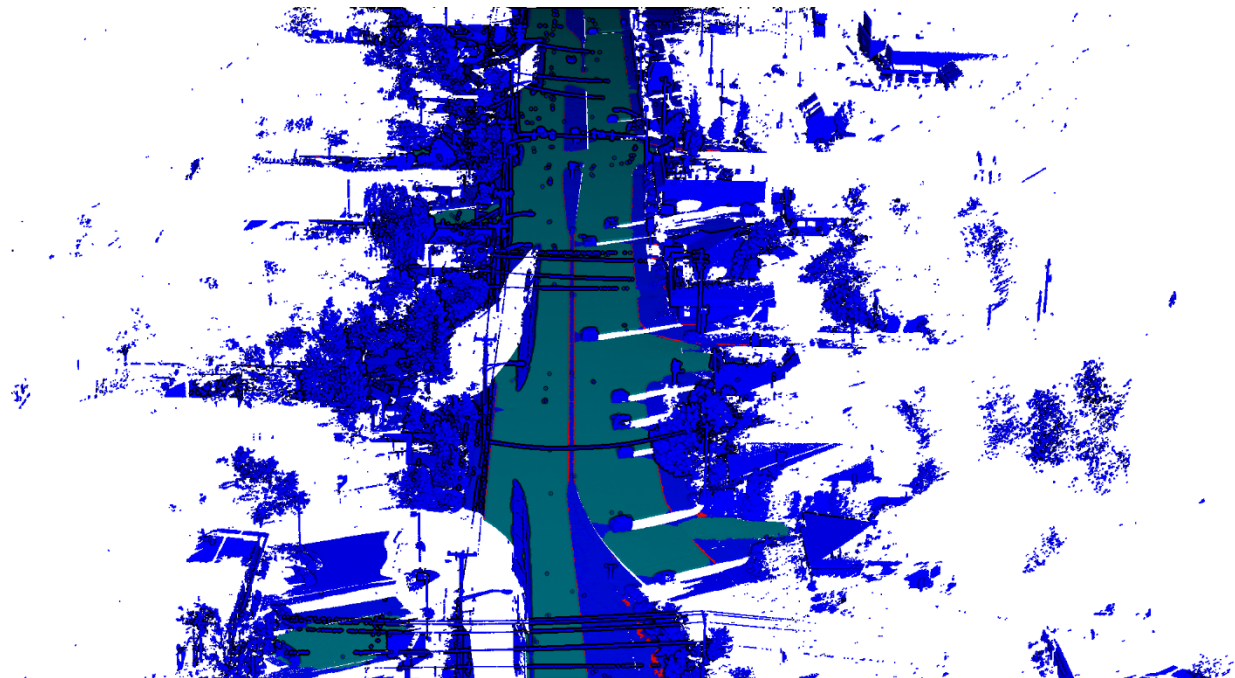


Figure 5-7. Example of the road surface segmentation on MLS dataset collected by ODOT using Leica Pegasus:Two where red and green points are labeled as curb and road surface, respectively.

6.0 FEATURE EXTRACTION ALGORITHMS

6.1 ROAD MARKING EXTRACTION

In this phase, road markings from point clouds are segmented using methods previously proposed by Jung et al. (Jung et al. 2019, 2020). MLS data usually contains a large number of points from different objects within the scene. The significant size of lidar point clouds often results in time-intensive processing due to heavy computational demands. To address this, the study adopts a 2D rasterization technique to decrease computational complexity. Initially, points that define the road surface are separated and rasterized onto a 2D image. This image indicates occupied pixels—those with at least one scan point—with a mean intensity value scaled from 0 to 255. Pixels without any scan points are assigned a value of 0.

For rasterization, a fixed cell size of 0.1 m is adopted. This not only ensures high efficiency for subsequent processing steps but is also necessary to maintain uniform image patch sizes for training the deep learning model, which classifies the bike symbols. To reduce the number of unoccupied pixels, roadway scan points are rotated around the z-axis to align with the primary road direction, using the trajectory recorded by the MLS inertial navigation system. Subsequently, a binary image is generated by applying an intensity threshold, which extracts the road markings based on their high radiometric contrast against the road pavement. The Otsu method (Otsu, 1979) is utilized to determine this threshold. Figure 6-1 shows an example of extracted road markings overlain on the point cloud data.

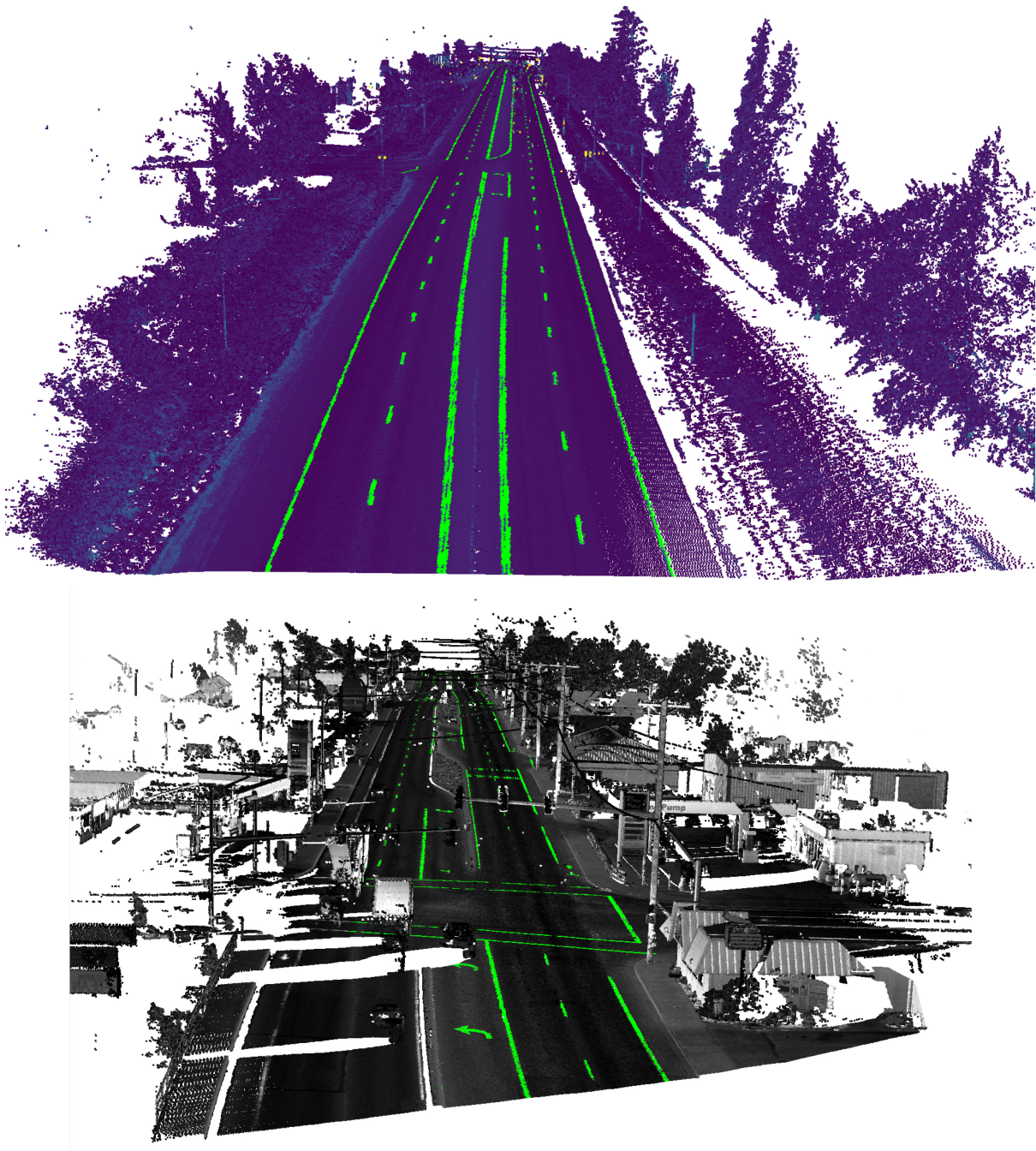


Figure 6-1. Example road marking extraction (green points) for two different highways.

6.2 ROAD MARKING CLUSTERING

Road marking clustering is an essential step to identify individual road markings. Road marking clustering isolates individual road markings from MLS data, allowing for instance-level classification. Initially, the connected component labeling (CCL) method is applied to isolate individual road markings with unique integer labels. However, since the CCL method can result

in under- or over-clustering, a series of refinement steps is required to address these issues. The proposed road marking clustering approach includes two steps to address the issues of under- and over-clustered road markings. In most cases, under-clustering occurs due to lane-type markings, such as lane and standard parallel markings, being connected to one another. These connections can be divided into junctions and corners: junctions are intersections where at least three lines connect, and corners are intersections where two lines connect. To efficiently identify the connections, a kernel search is applied for junction detection, followed by incremental line segmentation for corner detection. After dividing the under-clustered lane markings into smaller clusters, over-clustered lane markings are identified and merged through a rigorous rule-based approach, focusing on their proximity and alignment. All of the parameters required for the road marking clustering have been optimized using information from a traffic line manual (ODOT, 2023). A more detailed explanation of the clustering approach is documented in Jung et al. (In Press). Figure 6-2 compares the conventional CCL method and the proposed clustering approach, which successfully links portions of markings across gaps from worn markings. Figure 6-3 shows an additional example of correct linkage from gaps due to occlusions from vehicles. Figure 6-4 provides several examples of successful clustering in a variety of situations as well as some limitations such as connected markings, very large gaps, and high curvature.

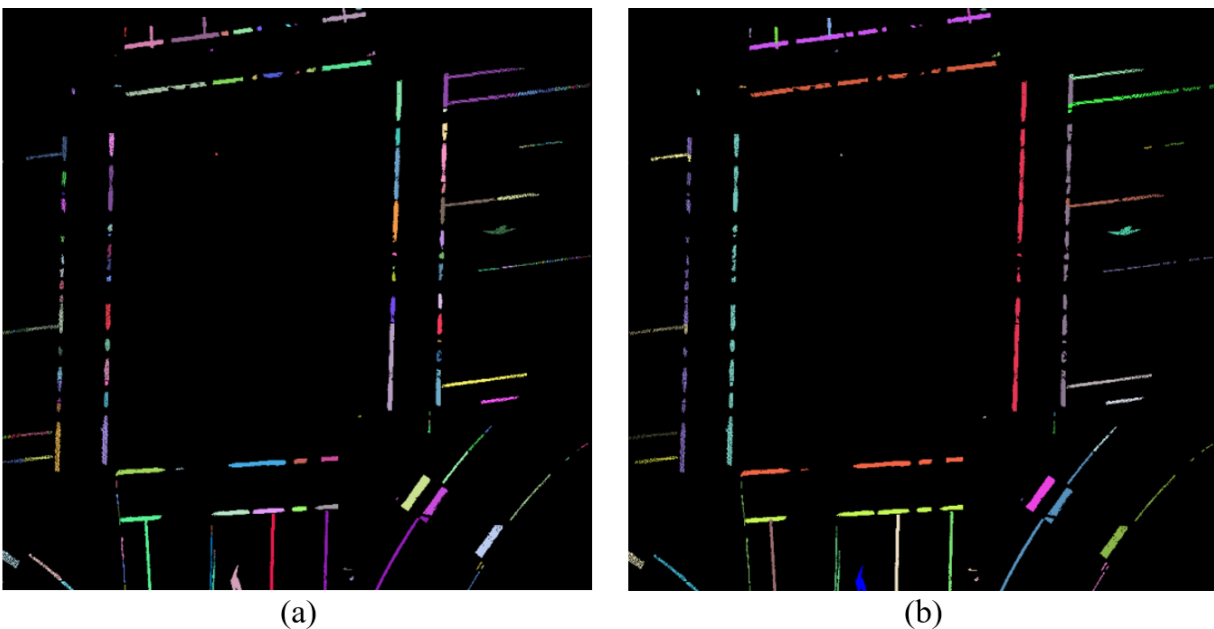


Figure 6-2. Example of the road marking clustering by: (a) connected component labeling method and (b) the proposed approach.

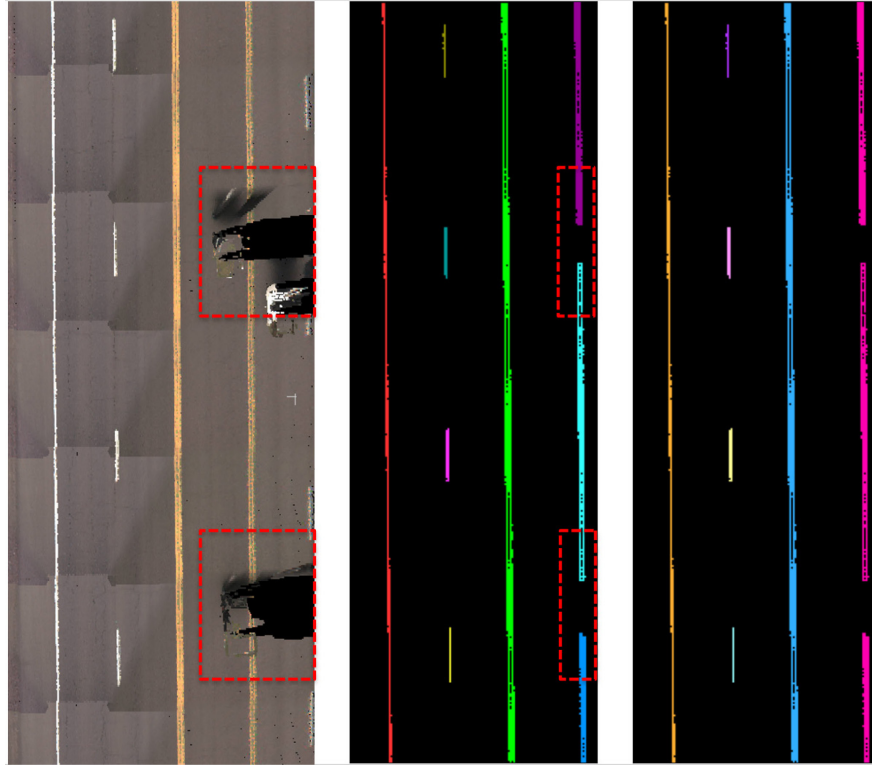


Figure 6-3. Example of the improvements provided by the road marking clustering algorithm. The left shows the point cloud colored by RGB with vehicles blocking the view of the scanner. The middle shows the initial clustering (each color identifies a unique marking) where the marking is separated at the data gaps. The right shows the corrected marking where the clusters belonging to the same marking are linked.



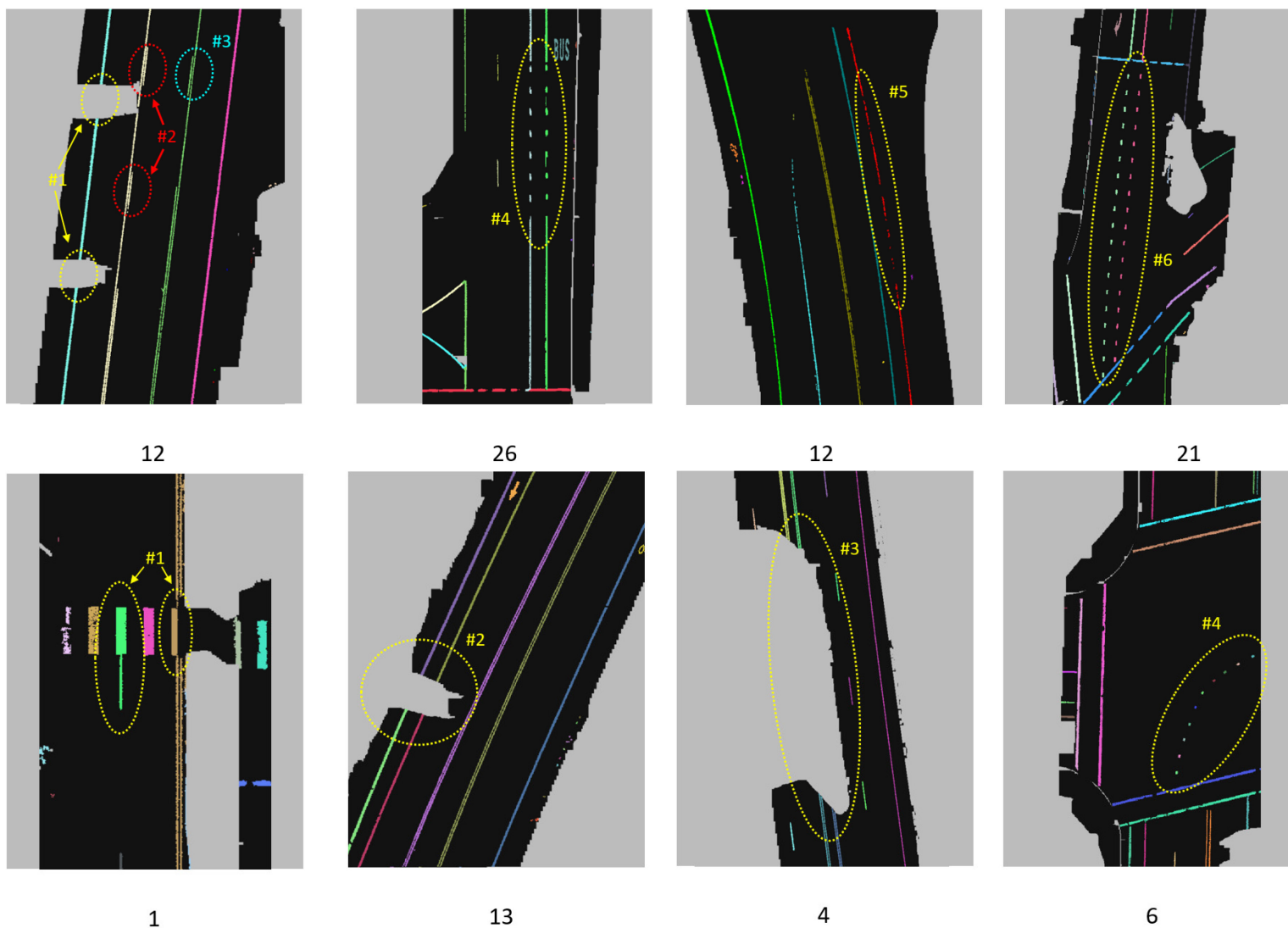


Figure 6-4. Examples of road marking clustering. The top row shows examples of successful linkage of markings across data gaps or dotted lines. The bottom row shows areas where the clustering algorithm was not successful due to connections between markings, (1), large data gaps (13 and 4), or high curvature or wear (6).

6.3 ROAD LANE EXTRACTION

In the previous step, once the road markings have been extracted and clustered, line fitting is applied to each individual marking. This process enables the determination of the length and width of each marking, which are used for identifying lane and dash markings. With these lane and dash markings, individual lanes on the road surface can be delineated by establishing the centerlines of each lane. Figure 6-5 illustrates a conceptual approach to identifying individual lanes. The algorithm requires prior classification of lane and dash markings (depicted as green lines in Figure 6-5) before lane extraction can proceed. The basic workflow for extracting the center lines consists of dividing the trajectory into smaller segments at regular intervals (which should be less than the length of the dash markings to ensure at least one intersection is detected within each dash marking, typically 2.5 m by default). For each trajectory segment, a line is fitted to calculate its parameters. A transverse line, perpendicular to the segment and intersecting at its midpoint, is then calculated. These transverse lines are generated every 2.5 m along the trajectory. The intersections of the transverse lines with lane or dash markings are calculated. The center points between intersections on the same transverse line are then determined and grouped by their orthogonal distances to the trajectory. This process identifies center points along the same lane, which are connected to form the center lines for each lane. By connecting these center points, center lines representing the individual lanes on the road surface are generated. The curb information is used to help identify the edge of the road.

In the case of dash markings, large gaps between dashes in the longitudinal direction can prevent the calculation of intersections with transverse lines. To address this, dash markings in the same longitudinal direction are identified and connected to create a continuous lane marking. This is achieved by performing line fitting on the points of individual dash markings using a least-squares method. The line parameters are then used to identify and connect dash markings aligned in the same longitudinal direction, as illustrated by the blue dotted line in Figure 6-5, forming an extended lane marking that allows for the calculation of intersections with transverse lines in the gaps. Figure 6-6 illustrates an example of the extracted road center lines to represent individual road lanes.

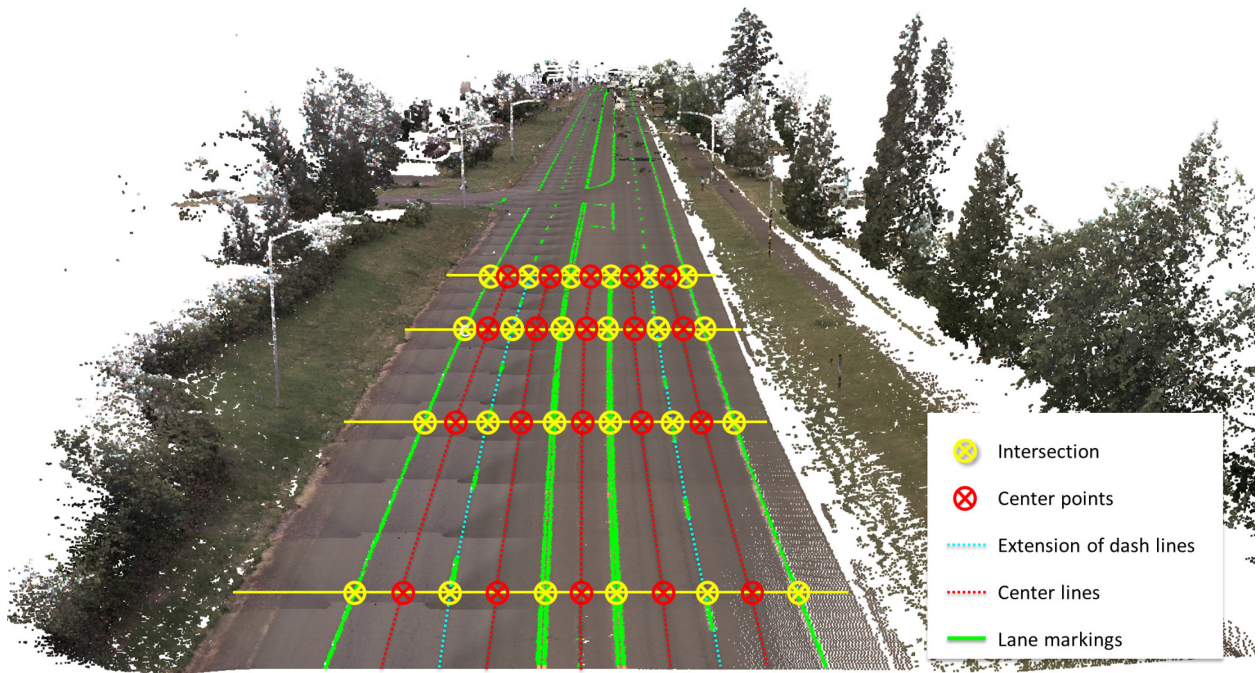


Figure 6-5. Conceptual image for identifying individual lanes using the extracted road markings.

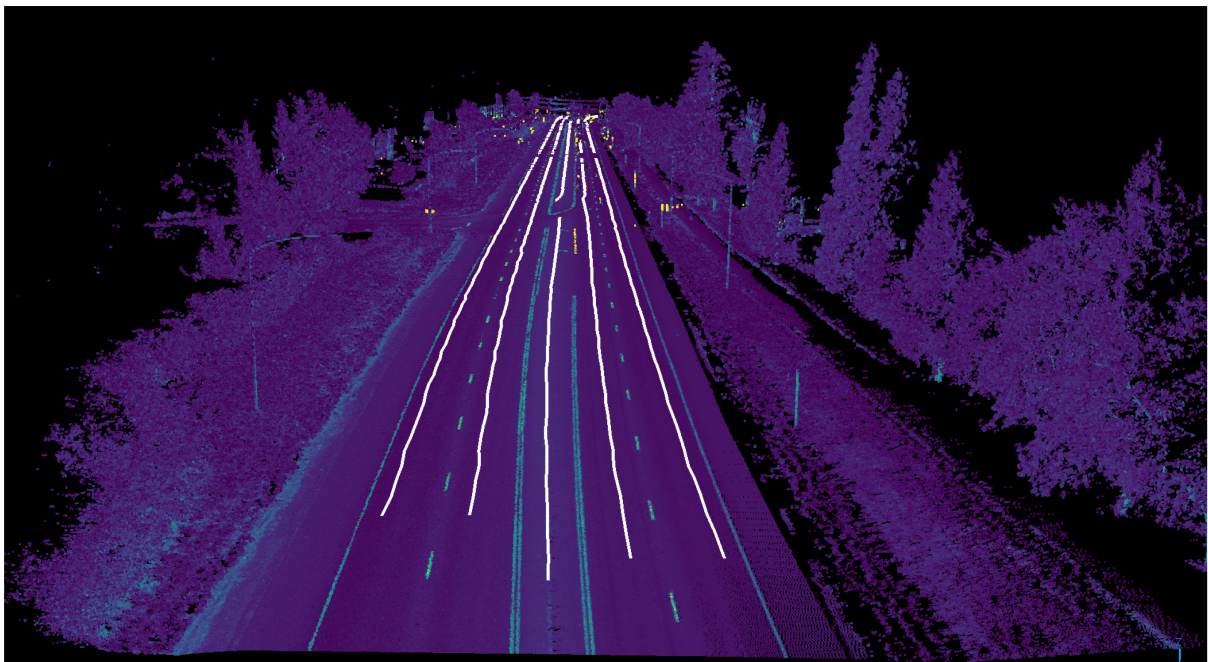


Figure 6-6. Example of road lane extraction (white lines): The height of the road lanes has been adjusted to match the vehicle trajectory height for visualization purposes.

6.4 RAISED MEDIAN AND STAGGERED CONTINENTAL EXTRACTION

The points that represent raised medians are typically eliminated during the ground filtering process, which leads to gaps in the rasterized road surface image. Consequently, potential raised medians can be identified by the voids that are enclosed by the road surface. However, occlusions might mistakenly be identified as raised medians. To eliminate false medians, two geometric criteria are enforced: 1) the length of the raised medians must exceed their width; and 2) the points delineated by the raised median should be elevated compared to the adjacent road surface. The figure illustrates a raised median as identified by the suggested method. Figure 6-7a presents point cloud data in RGB colors, while Figure 6-7b illustrates a segmented example of raised medians from the point cloud data.

To extract staggered continental markings (marked crosswalks), a Convolutional Neural Network (CNN) with a modified VGG6 architecture was used. However, the classification may miss some actual crosswalk markings due to wear. To identify worn markings, markings detected by the CNN method serve as seed markings. These seed markings are then clustered using Euclidean distance, focusing on the endpoints (x and y coordinates) of the line fitted to each marking. If the endpoints are within a specified distance threshold, corresponding to the roadway's width, they are grouped into a single cluster. In clusters with more than one crosswalk marking, further analysis is performed to detect any undetected crosswalk markings near the seed markers. This involves calculating the midpoint of each line fitted to the seed markings, then deriving a new line parameter through least squares adjustment based on these midpoints. If any markings intersect with this new line parameter, they are classified as marked crosswalk. Finally, any markings exceeding the specified length of 9 feet, as defined in the traffic line manual, are categorized as lane markings and excluded from the marked crosswalk markings. Figure 6-7b includes an example of a marked crosswalk segmented from the point cloud data.

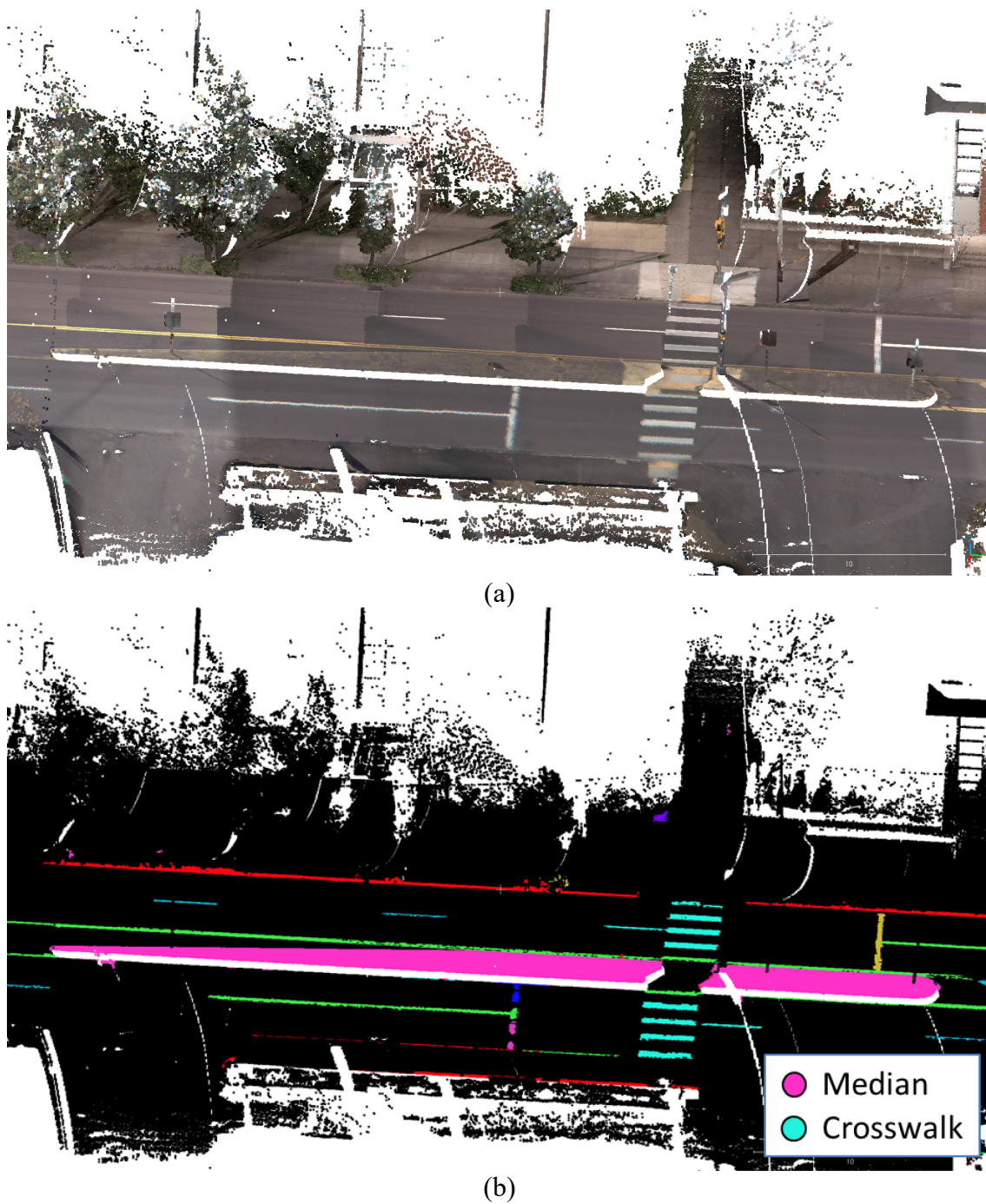
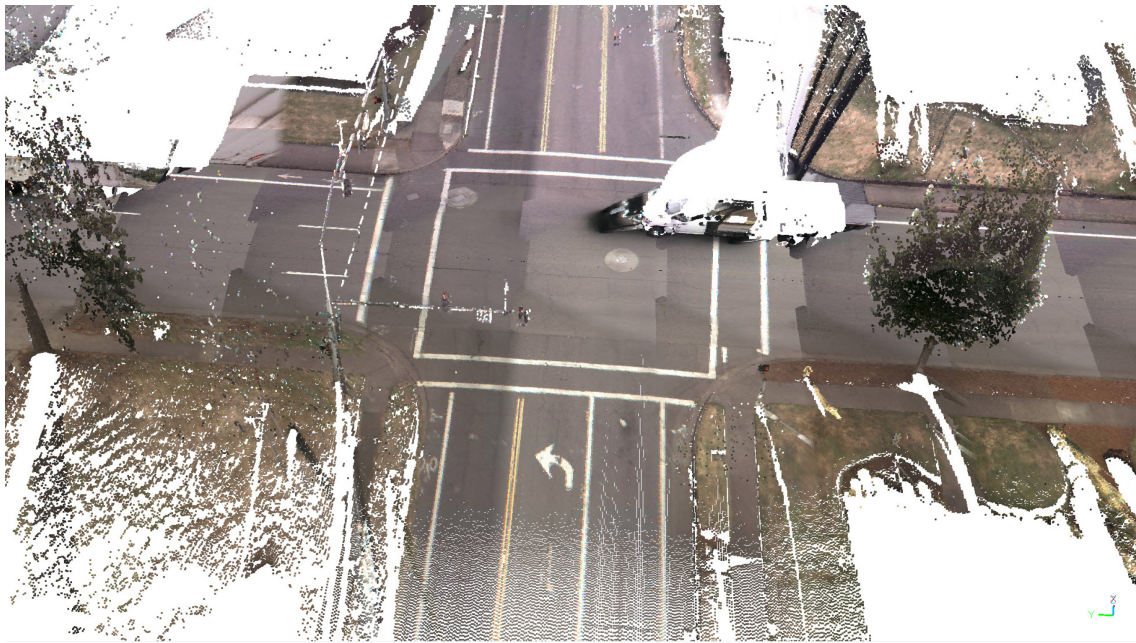


Figure 6-7. Example of raised median and marked crosswalk extraction: (a) the point cloud in RGB colors and (b) the extracted features in discrete colors.

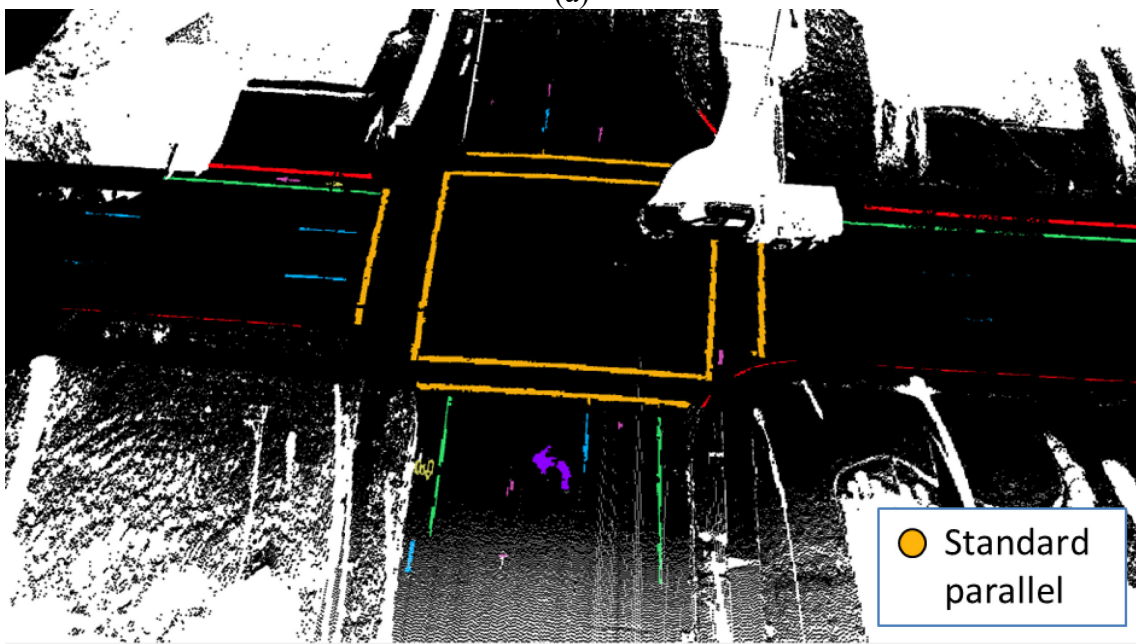
6.5 STANDARD PARALLEL EXTRACTION

The algorithm detects standard parallel markings (stop lines) in two orientations: transverse and longitudinal. To identify transverse stop lines, vehicle trajectory data are utilized. Unlike other lane markings, transverse stop lines span across the roadway, enabling their detection by calculating intersections with vehicle trajectory data. Only those markings intersecting with the trajectory and shorter than the roadway's width are considered as transverse stop lines.

Following this, longitudinal stop lines are identified, focusing only on those at intersections. The detection process starts with selecting lane markings and evaluating them. Pre-detected transverse stop lines are initially grouped using the Euclidean distance clustering method, based on the endpoints of line parameters fitted to each line. The distance threshold is set to three times the length of the longest transverse stop line to ensure that transverse lines at the same intersection are grouped together. Subsequently, the algorithm identifies the maximum and minimum y-coordinates within each cluster. Lane markings with endpoints within this y-coordinate range are identified as candidate longitudinal stop lines. Their orientations are then evaluated. Since transverse stop lines at the same intersection share a common orientation, any transverse line can be selected to check whether the candidate longitudinal stop lines are perpendicular to it. Lane markings whose orientation deviates by more than 45 degrees from the chosen transverse line are classified as longitudinal stop lines. Figure 6-8a presents point cloud data in RGB colors, and Figure 6-8b illustrates an example of stop lines segmented from the point cloud data.



(a)



(b)

Figure 6-8. Illustration of standard parallel markings: (a) the point cloud data in RGB colors and (b) the segmented markings highlighted in yellow.

6.6 BIKE LANE EXTRACTION

To distinguish bike lanes from other lanes detected in Section 3.3, two key constraints are applied: (1) a centerline that either passes over a bike symbol or (2) a lane situated near the curb is classified as a bike lane. Using the line parameters fitted to each marking, markings that are equal to or smaller than the bike symbol's length (24 inches) and width (40 inches) are selected as candidates according to the specifications in the traffic line manual. A modified VGG6 deep learning model was employed for bike symbol detection.

A centerline that lies sufficiently close to an adjacent curb is identified to represent a bike lane. This is logical since bike lanes are typically positioned along the edges of roadways, adjacent to curbs, to minimize disruption to traffic flow. The classification criteria utilize a proximity threshold set at half the standard bike lane width of 6 feet (1.8 meters), as outlined in the ODOT traffic line manual (2023). This equates to 3 feet (0.9 meters) for assessing the proximity of centerlines to curbs. However, in practice, to reduce the undetected bike lanes (false negatives), a conservative threshold of 1.2 meters was used.

Utilizing curb proximity to detect bike lanes can still mistakenly identify lanes near raised medians as bike lanes (false positives) because raised medians typically have curbs. To reduce such errors, any bike lanes adjacent to the medians, which are detected in Section 3.4, are excluded. This is achieved in the rasterized image by removing bike lanes that intersect with medians expanded through morphological dilation process using a structural element half the lane width (3.6 meters), thereby refining the selection of candidate bike lanes. Figure 6-9 shows an example of a bike lane extraction.

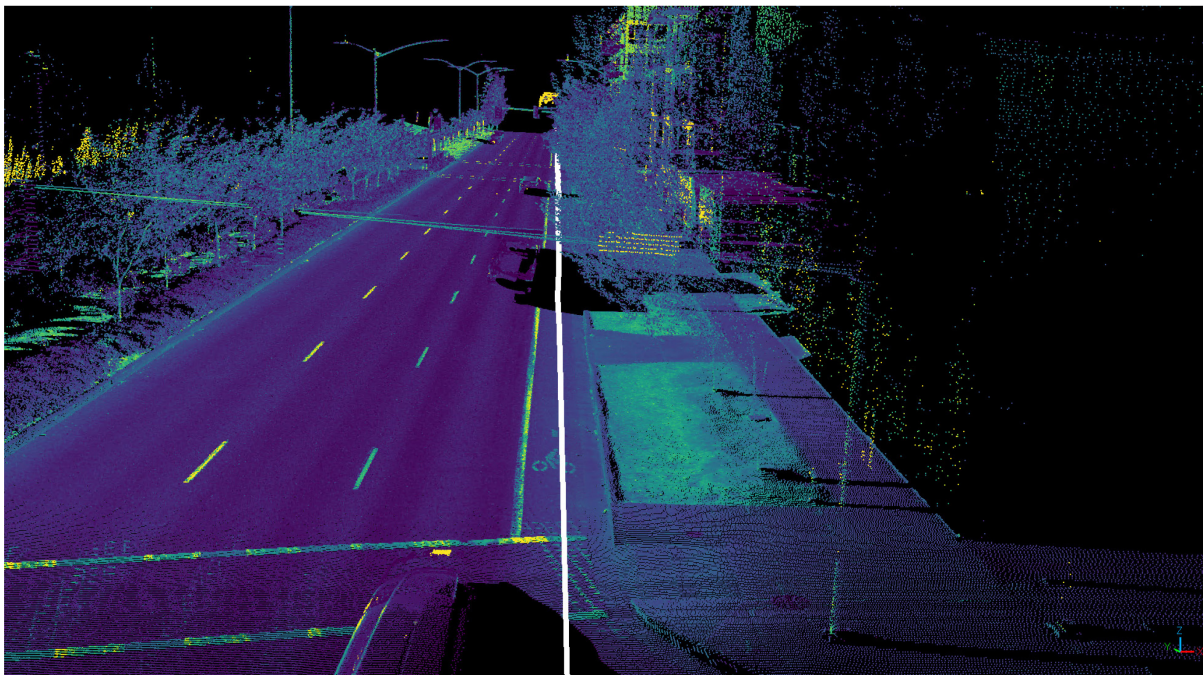


Figure 6-9. Example of bike lane extraction (white line): The height of the bike lanes has been adjusted to match the vehicle trajectory height for visualization purposes.

7.0 TESTING RESULTS

The algorithms were tested on several of the diverse datasets compiled in Section 3.2 to ensure that they are reliable and function for the most common situations throughout Oregon. The testing strategy consisted of the following:

1. Visual verification compared with layers in TransGIS.
2. Quantitative evaluation of the extracted objects in comparison with manually digitized features.

Table 7-1. Summary of the primary datasets used for the testing and evaluation.

Site	Highway	Dataset ID	Length (m)	Number of Points (Millions)
1	221	13	1437.43	18
2	99W Southbound	23	422.06	17
3	29	15	499.37	19
4	99E	32	967.23	18
5	213 Southbound	3	0.00	19
5	213 Southbound	2	23.20	19
Overall			3349.29	110

7.1 TRANSGIS COMPARISON

The output data layers from the ROME tool were visually compared to layers in TransGIS and recent statewide aerial imagery layers for a general assessment of performance. Several benefits to the ROME tool extraction were noted during the comparison:

1. The georeferencing accuracy of the ROME results was much improved in comparison to the TransGIS layers, which were not as precisely digitized, typically deviating by several meters.
2. TransGIS does not include cross walks from intersecting streets, which in many cases were successfully extracted from the ROME tool. Those that were not classified correctly with ROME were still extracted as markings and straightforward to reclassify in GIS software.
3. Curbs and pedestrian medians are not available in TransGIS. (There is a related layer showing the location of barriers).
4. Supporting road marking information is also not currently available in TransGIS but can be provided by the ROME tool.

7.2 QUALITATIVE EVALUATION

To quantitatively evaluate the effectiveness of the tool in object extraction, the following procedure was employed for bike lanes, crosswalks, and medians:

1. Objects were manually digitized by the research team in Civil 3D software from the datasets identified in Table 7-1 encompassing each of the five sites. Note that the extraction was only completed in areas with mobile lidar data of reasonable quality (e.g., minimal occlusions, near the trajectory).
 - a. Bike lanes were extracted as polyline features in the center of the bike lane. They were not digitized to continue through the intersections.
 - b. For standard parallel crosswalks, each painted line (e.g., two lines per side of intersection) were manually digitized as separate polyline features.
 - c. For staggered continental crosswalks, each block was digitized as a single line.
2. These manually extracted objects were then used as reference data and spatially compared with the ROME outputs to identify:
 - a. *True positives (TP)*: locations where both the ROME tool and the manual extraction indicated the presence of the same objects.
 - b. *False Positives (FP)*: locations where the ROME tool indicates the presence of a feature that are not actually present.
 - c. *False Negatives (FN)*: locations where the ROME tool did not successfully identify the presence of a feature that are present and captured in the mobile lidar data.
 - d. Note that *True Negatives (TN)* were not computed or considered because they would artificially weight the performance higher, especially for crosswalks given that the length of roadway without crosswalks is substantially more than the area with crosswalks.
3. The lengths of these feature segments in each category were computed.
4. Several performance metrics were then computed from the lengths of those features for each category (e.g., bike lanes, and crosswalks) separately. These metrics all range from 0 to 100%.
 - a. *Precision*: $Pr = TP / (TP + FP)$. Precision is a measure of the correctly labeled segment lengths divided by the total segment lengths that were labeled as the object. High values indicate good precision.

- b. *Recall*: $Re = TP / (TP + FN)$. Recall is a measure of the correctly labeled segment lengths divided by the total available lengths of that feature in the reference data and should have been labeled as the object. High values indicate good recall rates.
 - c. *Omission Error*: $FN / (TP + FN)$. Omission Error relates the length of polylines of the object missed divided by the total segment lengths that should have been labeled as the object. Low values indicate good omission error rates.
 - d. *Commission Error*: $FP / (TP + FP)$. Commission error relates the length of polylines that were incorrectly labeled as the object divided by the total segment lengths that should have been labeled as the object. Low values indicate good commission error rates.
 - e. *F1-score* = $2 * (Pr * Re) / (Pr + Re)$. The F1-score is a measure of the predictive performance and balances precision and recall rates.
5. The time required was recorded for key stages of the operation. Note that the runtime of the tool is an automated portion that does not require interaction from the user. Hence, the process is highly efficient where data can be batch processed:
 - a. The time required for running the curb extraction script. (Both setup and runtime)
 - b. The time required for the ROME tool. (Both setup and runtime)
 - c. An estimate of the time required to clean up the output datasets.
 - d. The time required to manually digitize the features.
6. For the time associated with the tools, results are normalized to length of the section (km) and the number of points in order to determine the scalability of the tools to process multiple datasets in batch mode.

7.3 TESTING RESULTS

7.3.1 Bike lanes

Table 7-2 shows the performance results of the algorithm for extracting bike lanes. Example extraction results are shown in Figure 7-1 to Figure 7-5. In the evaluation, the following was noted regarding the performance of the algorithm.

- Several false positives were observed as bike lanes crossing through the intersection. These false positives were logical since bikes would continue through the intersection despite the bike lane not being marked in the intersection itself. ROME has a post processing step in which it connects the bike segments across intersection. (Dataset 13)
- For Dataset 23, the bike lanes were extracted as a vehicle lane; however, they were not extracted when ROME was run in 'bike lane' only mode.

- For Dataset 15, ROME was successful in capturing bike lanes in some of the occluded areas.
- For Dataset 32, part of the extracted bike lanes were on sidewalks due to multiple curb sets. These were all still assumed to be true positives for the evaluation.
- There were no bike lanes in Site 3. ROME did not extract any bike lanes (false positives) here, which is the correct result.
- For Dataset 2, there was only a small segment of bike lane present, located on a crossroad in ROME. This was not captured in ROME as the bike lanes need to be parallel to the trajectory direction for classification.
- Overall, the bike lane extraction performed well with an overall F1-score of 80.58%. The omission errors were higher than the commission errors.

Table 7-2. Summary performance metrics for bike lane extraction and observations from the qualitative analysis.

Site	Dataset ID	Length Ref (m)	Length Extracted (m)	TP (m)	FP (m)	FN (m)	Prec.	Recall	Om. Error	Com. Error	F1-Score
1	13	1437.43	1497.70	1341.56	156.14	95.87	89.57%	93.33%	6.67%	10.43%	91.41%
2	23	422.06	0.00	0.00	0.00	422.06	0.00%	0.00%	100.00%	0.00%	0.00%
3	15	499.37	399.75	353.40	46.35	145.97	88.41%	70.77%	29.23%	11.59%	78.61%
4	32	967.23	722.95	710.10	12.85	257.13	98.22%	73.42%	26.58%	1.78%	84.03%
5	3	0.00	0.00	0.00	0.00	0.00	N/A	N/A	N/A	N/A	N/A
5	2	23.20	0.00	0.00	0.00	23.20	0.00%	0.00%	100.00%	0.00%	0.00%
Overall		3349.29	2620.40	2405.06	215.34	944.23	91.78%	71.81%	28.19%	8.22%	80.58%

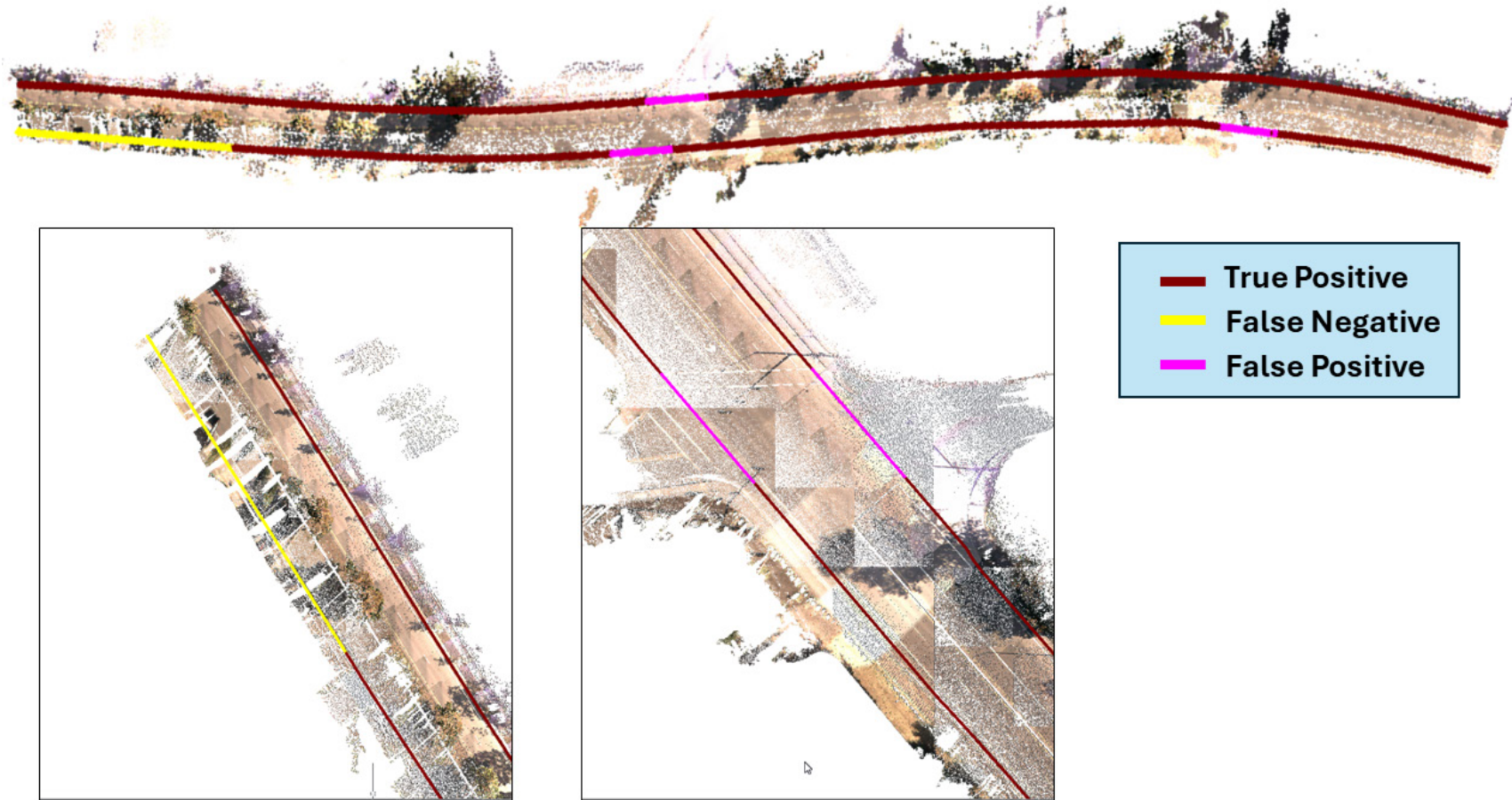


Figure 7-1. Bike lane extraction results for the evaluation section selected from Dataset 13 showing true positives, false negatives, and false positives. In this section, the false positives tend to correspond with intersections and are connections between the bike lane segments. On the left side a small portion of the bike lane is missed; however, it is on the opposite side of the road from the mobile lidar trajectory where the data were sparse due to occlusions.

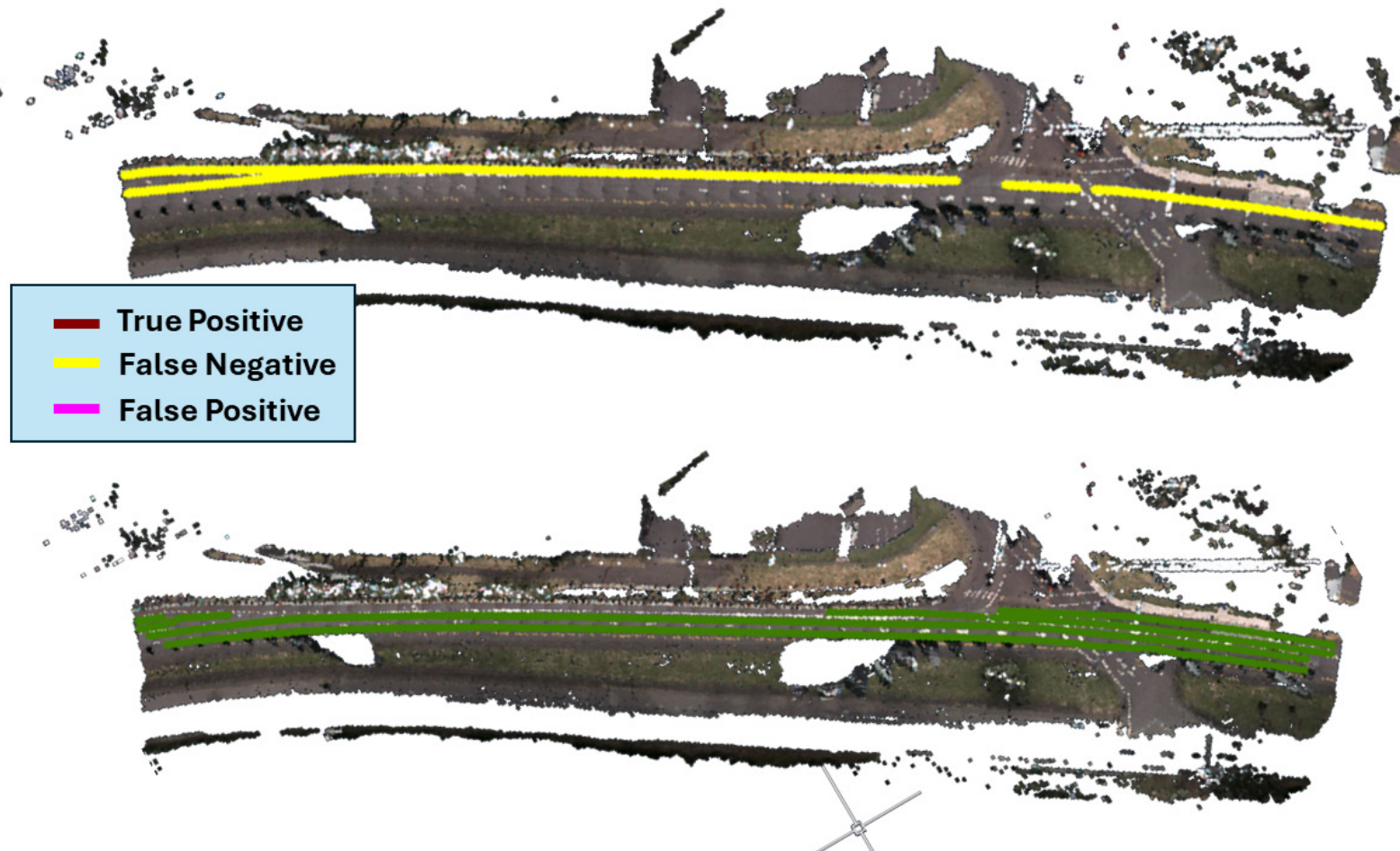


Figure 7-2. Bike lane extraction results for the evaluation section selected from Dataset 23 showing true positives, false negatives, and false positives. In this section, the entire bike lane was missed. It was, however, extracted as a vehicle travel lane represented as green polylines.

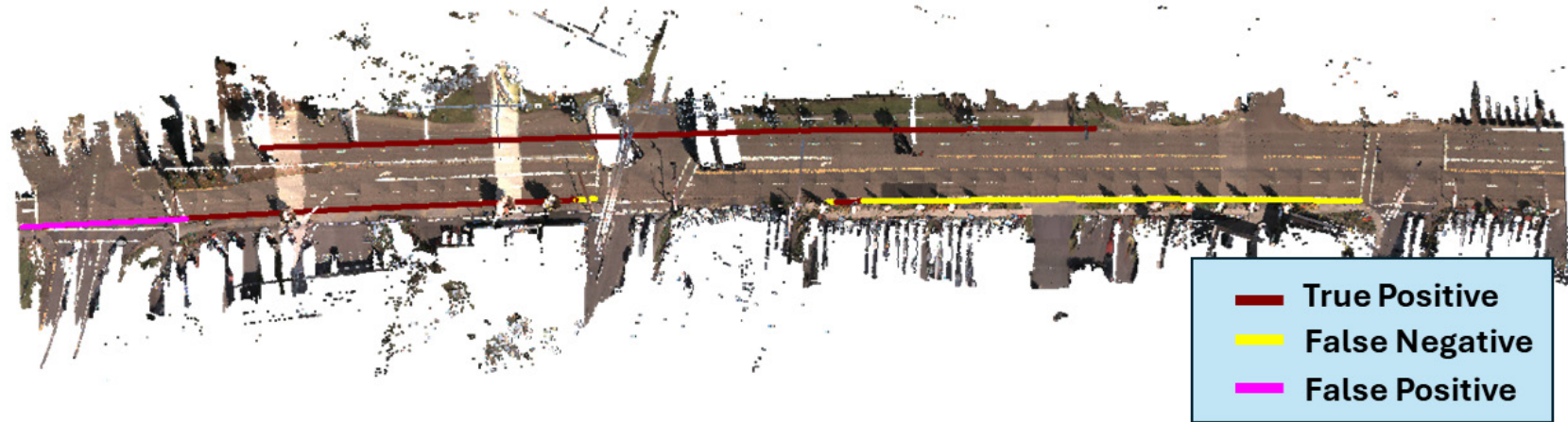


Figure 7-3. Bike lane extraction results for the evaluation section selected from Dataset 15 showing true positives, false negatives, and false positives. ROME has successfully extracted a bike lane in the occluded area (no data) using its post-processing mechanism.

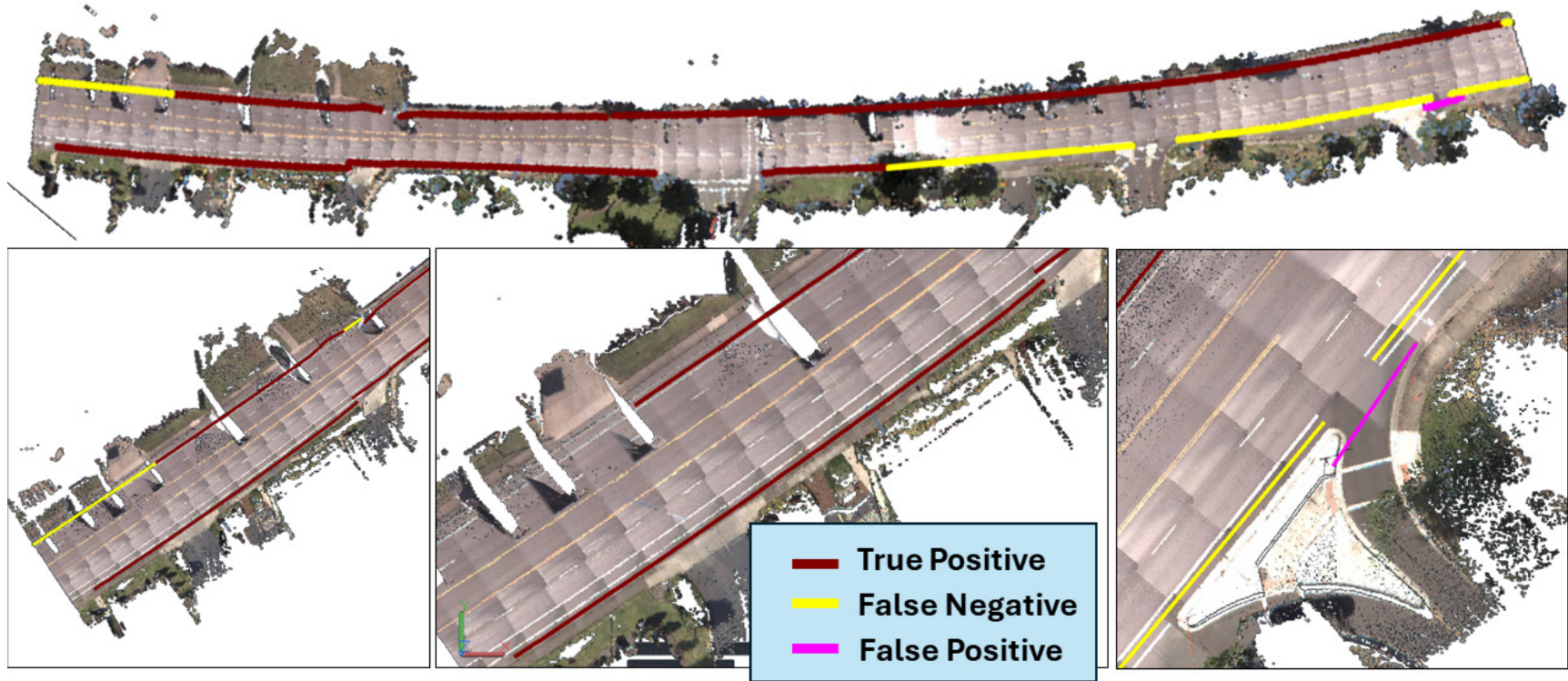


Figure 7-4. Bike lane extraction results for the evaluation section selected from Dataset 32 showing true positives, false negatives, and false positives. The false negatives on the left are a result of occlusions from vehicles. The false negatives on the right are likely due to multiple curbs extracted.

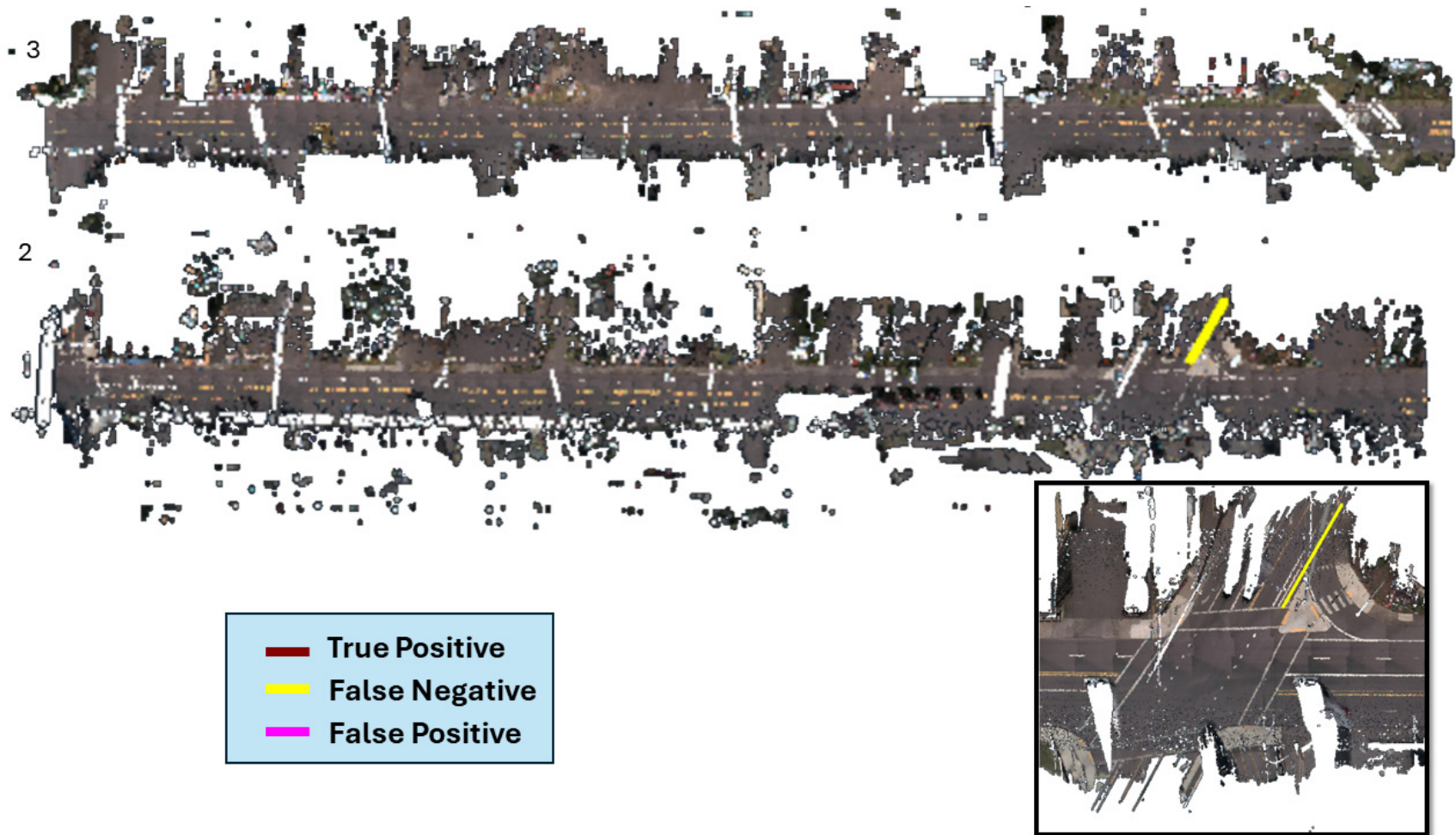


Figure 7-5. Bike lane extraction results for the evaluation section selected from Datasets 2 and 3 showing true positives, false negatives, and false positives. The bike lane on the crossroad was missed because ROME only extracts bike lanes parallel to the trajectory.

7.3.2 Crosswalks

Table 7-3 summarizes the evaluation results for the crosswalks. Overall, good performance is achieved with a F1-score of 73.46%. The omission errors were substantially higher than the commission errors. Figure 7-6 to Figure 7-11 show examples of the crosswalk extraction. For Dataset 13, most of the missed crosswalks located on the other side of the road which has a lower density points. They would likely be detected after processing acquired data from the other side of the road. For Dataset 23, the algorithm successfully extracted crosswalks that were at 45-degree angles to the trajectory in addition to those parallel and perpendicular to the crosswalk. In many false negative cases, ROME successfully extracted a marking at the location of the crosswalk; however, the labeling classification was incorrect. Hence, these do not need to be digitized, but can be simply relabeled to correct the output. Crosswalks near medians or going through one lane tend to be more challenging for extraction due to the smaller dimension (e.g., Dataset 11).

Table 7-3. Summary performance metrics for crosswalk extraction and observations from the qualitative analysis.

Site	DS#	Length Ref. (m)	Ext. Length (m)	TP (m)	FP (m)	FN (m)	Prec.	Re.	Om. Error	Comm. Error	F1- Score
1	13	144.25	64.51	64.51	0.00	79.74	100.00%	44.72%	55.28%	0.00%	61.80%
2	23	138.26	100.79	93.49	7.30	44.77	92.76%	67.62%	32.38%	7.24%	78.22%
3	15	184.83	200.98	138.90	62.08	45.93	69.11%	75.15%	24.85%	30.89%	72.00%
4	32	137.86	99.66	99.66	0.00	38.20	100.00%	72.29%	27.71%	0.00%	83.92%
5	3	69.88	7.00	7.00	0.00	62.88	100.00%	10.02%	89.98%	0.00%	18.21%
5	2	269.89	191.79	187.66	4.13	82.23	97.85%	69.53%	30.47%	2.15%	81.29%
Overall		944.97	664.73	591.22	73.51	353.75	88.94%	62.56%	37.44%	11.06%	73.46%

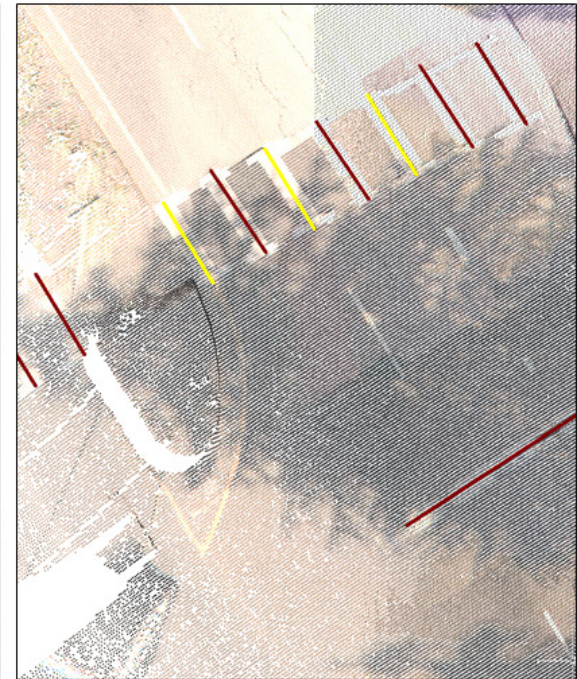
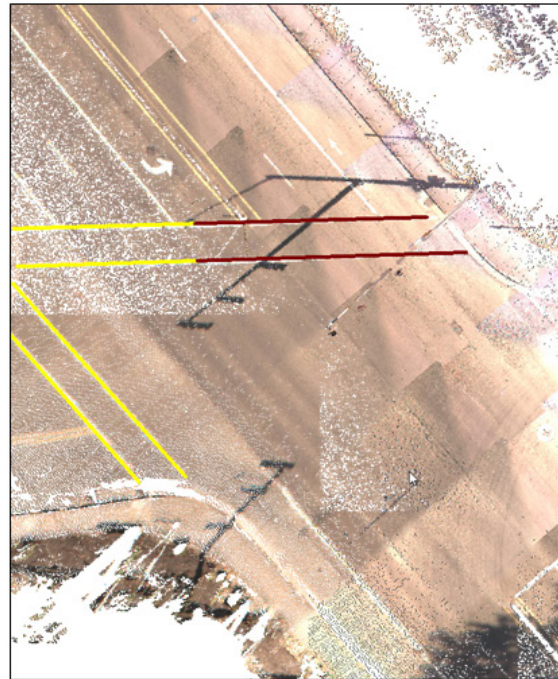
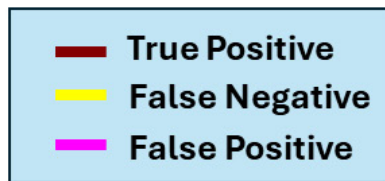


Figure 7-6. Crosswalk extraction results for the evaluation section selected from Dataset 13 showing true positives, false negatives, and false positives of the bike lane. While portions of the crosswalks are missing, ROME still identifies the presence of a crosswalk.

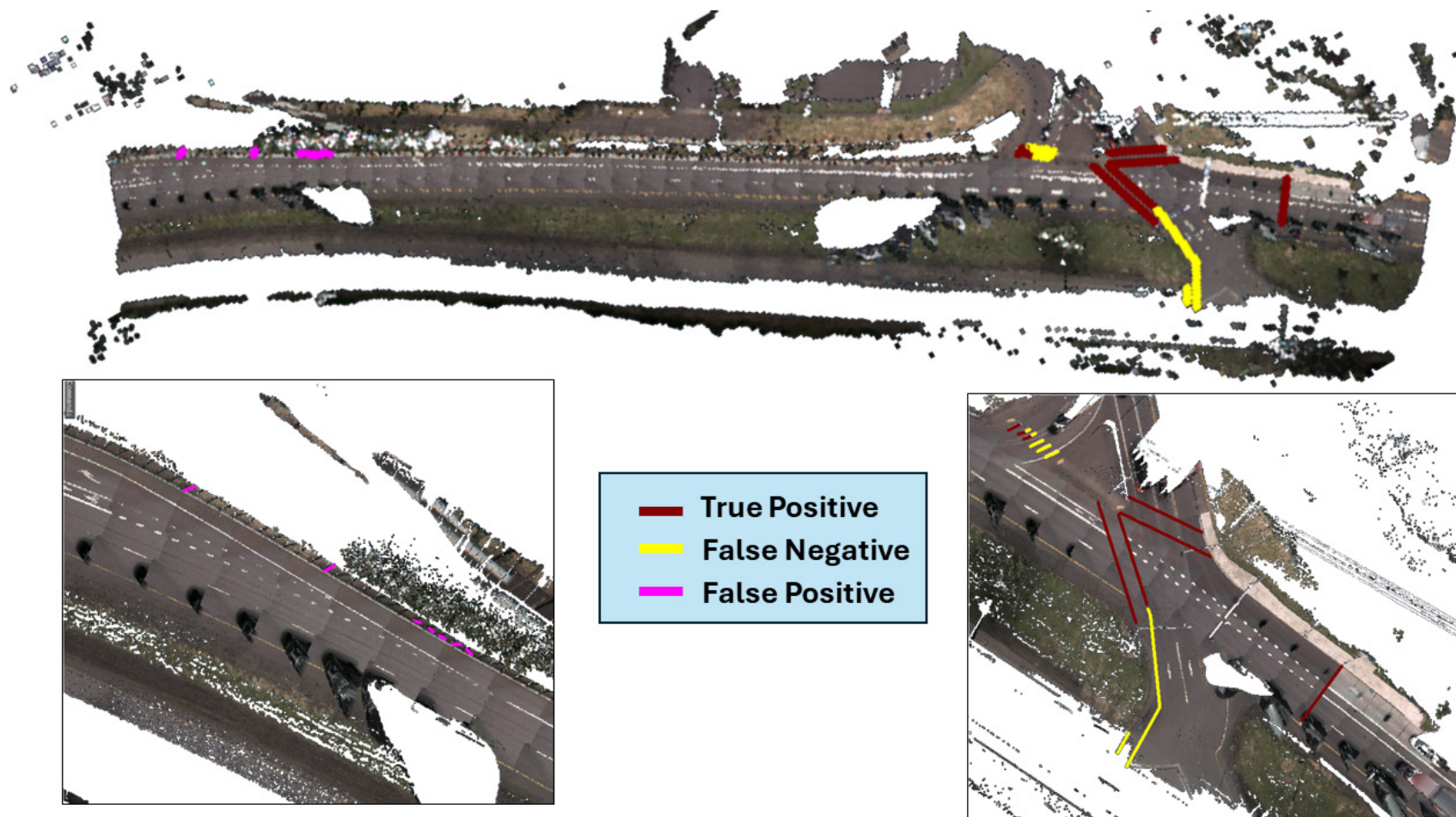


Figure 7-7. Crosswalk extraction results for the evaluation section selected from Dataset 23 showing true positives, false negatives, and false positives. Some false positives occur in a noisy section of the data on rumble strips. Most crosswalks are extracted; however a portion of the cross walk near the median was missed.

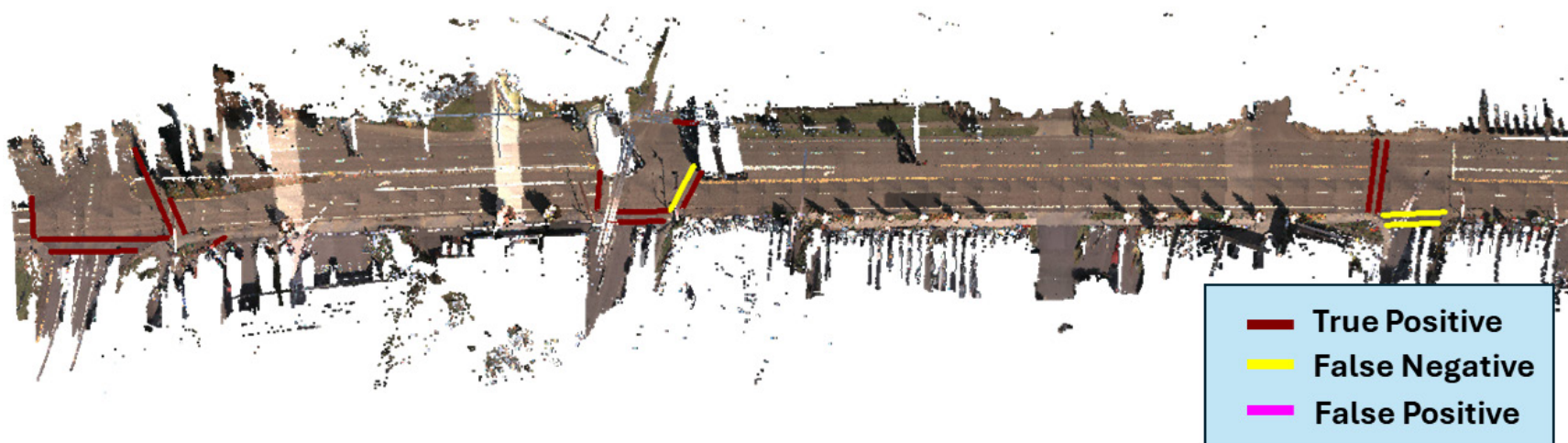


Figure 7-8. Crosswalk extraction results for the evaluation section selected from Dataset 15 showing true positives, false negatives, and false positives. Most portions of the crosswalk were correctly identified. Most false negatives were extracted as lane markings but not properly classified as crosswalks.

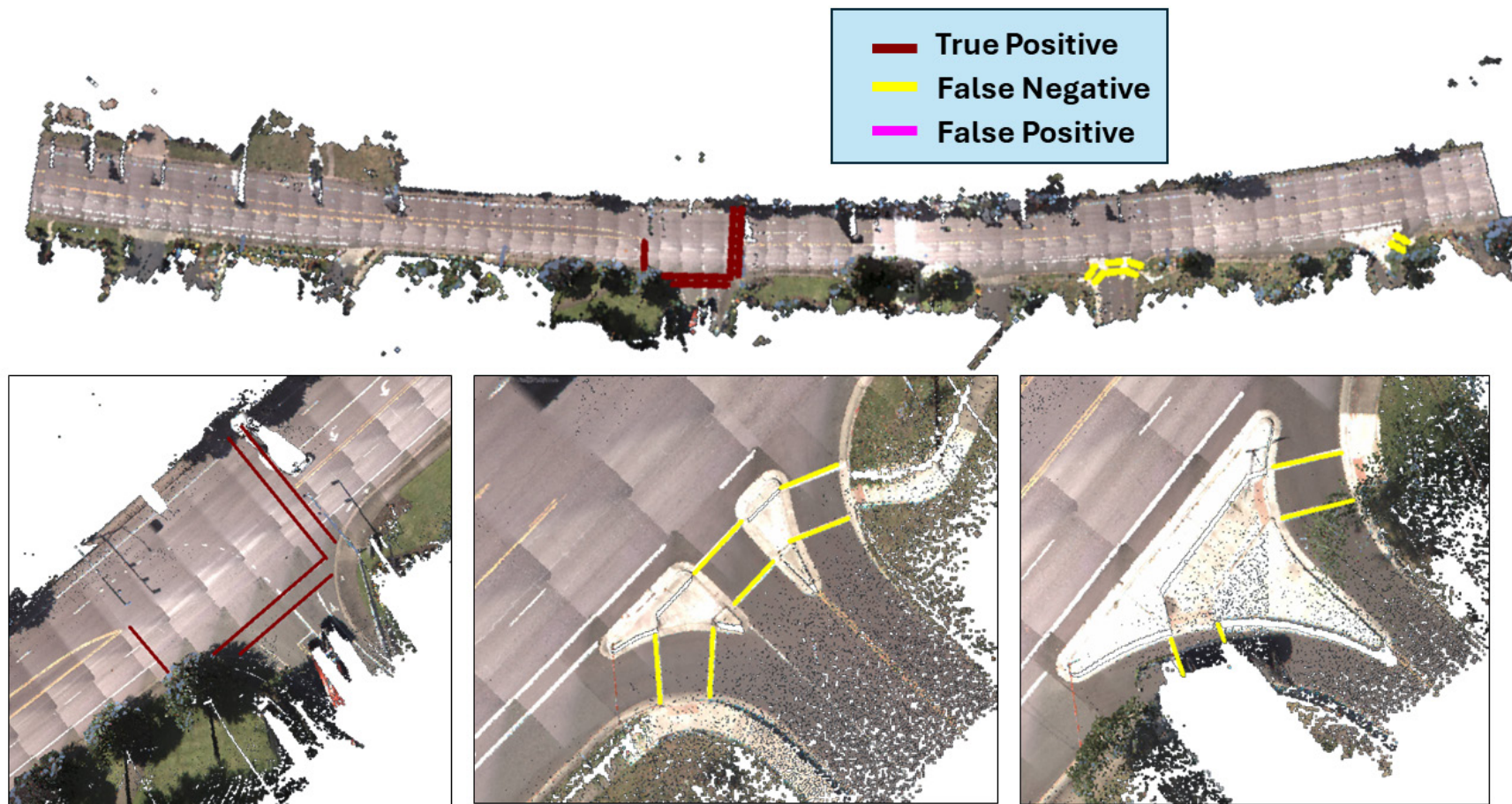


Figure 7-9. Crosswalk extraction results for the evaluation section selected from Dataset 11 showing true positives, false negatives, and false positives. ROME struggled to extract the smaller crosswalk sections near the medians.

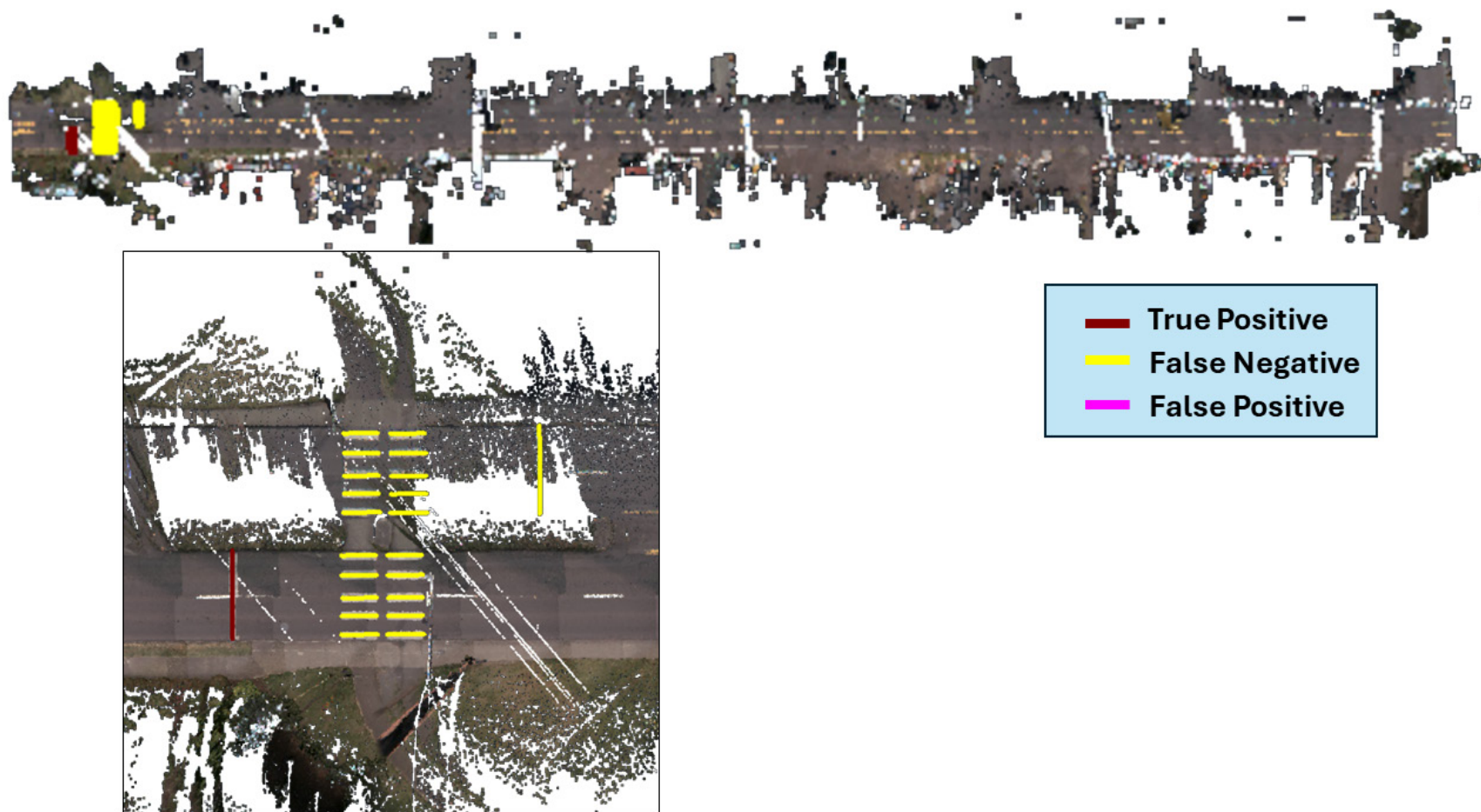


Figure 7-10. Crosswalk extraction results for the evaluation section selected from Dataset 3 showing true positives, false negatives, and false positives.

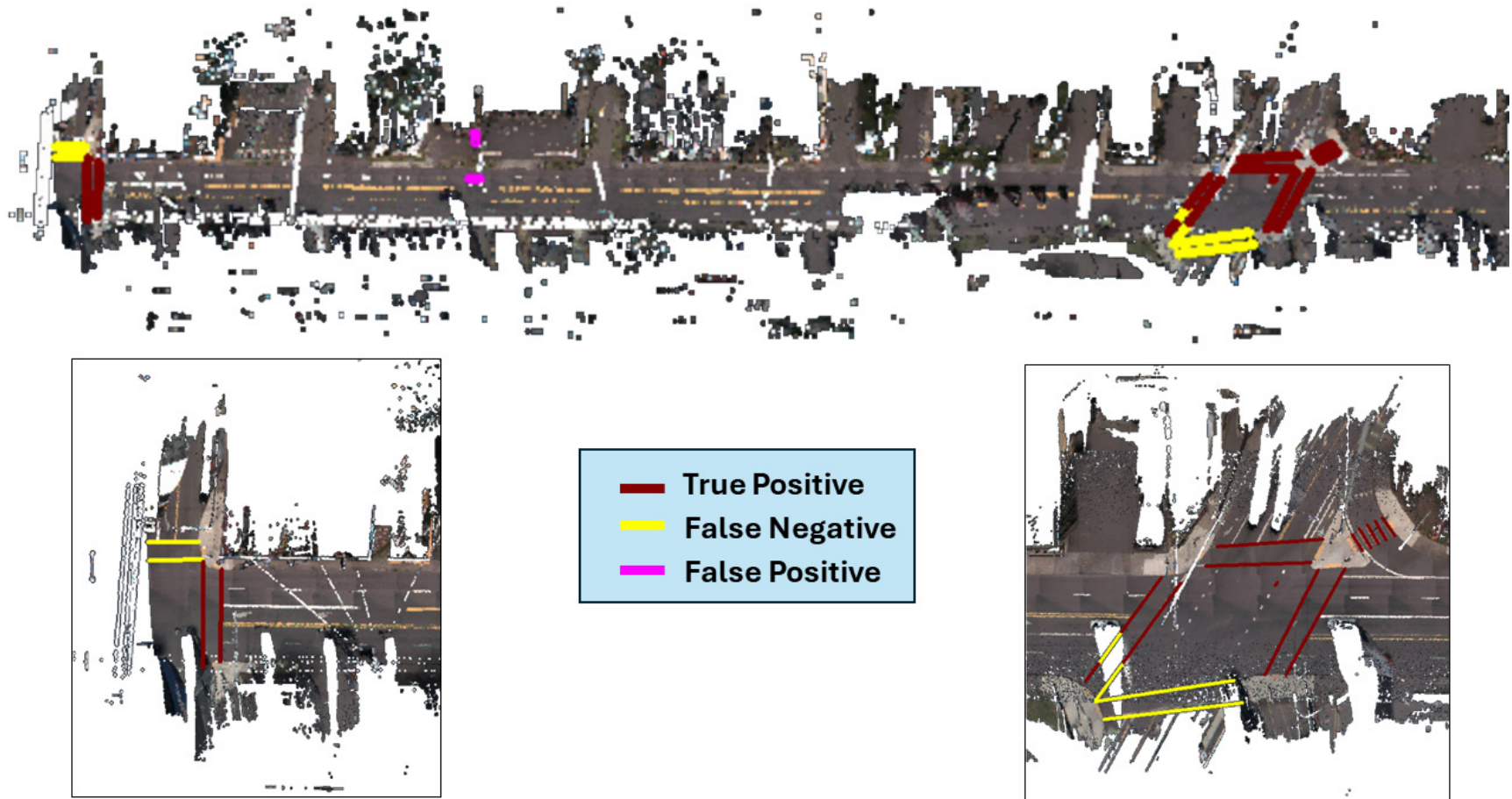


Figure 7-11. Crosswalk extraction results for the evaluation section selected from Dataset 2 showing true positives, false negatives, and false positives.

7.3.3 Medians

The median extraction proved challenging for the ROME software. Table 7-4 summarizes the evaluation results. Figure 7-12 to Figure 7-14 show example results of the extraction. It is recommended that medians be extracted manually using the ROME outputs and curb outputs as a guide. The curb extraction tends to show clear definition of the medians; hence the operator simply needs to select and reclassify those sections as medians rather than digitize them.

Table 7-4. Summary performance metrics for median extraction and observations from the qualitative analysis.

Site	Dataset ID	Length Ref. (m)	Length Extracted (m)	TP (m)	FP (m)	FN (m)	Precision (%)	Recall (%)	Omission Error (%)	Commission Error (m)	F1-Score
1	13	0.00	0.00	0.00	0.00	0.00	N/A	N/A	N/A	N/A	N/A
2	23	15.21	0.00	0.00	0.00	15.21	0.00%	0.00%	100.00%	0.00%	0.00%
3	15	55.20	62.03	0.00	62.03	55.20	0.00%	0.00%	100.00%	100.00%	0.00%
4	32	31.45	10.32	10.32	0.00	21.13	100.00%	32.81%	67.19%	0.00%	49.41%
5	3	0.00	0.00	0.00	0.00	0.00	N/A	N/A	N/A	N/A	N/A
5	2	13.84	6.70	6.70	0.00	7.14	100.00%	48.41%	51.59%	0.00%	65.24%
Overall		115.70	79.05	17.02	62.03	98.68	21.53%	14.71%	85.29%	78.47%	17.48%

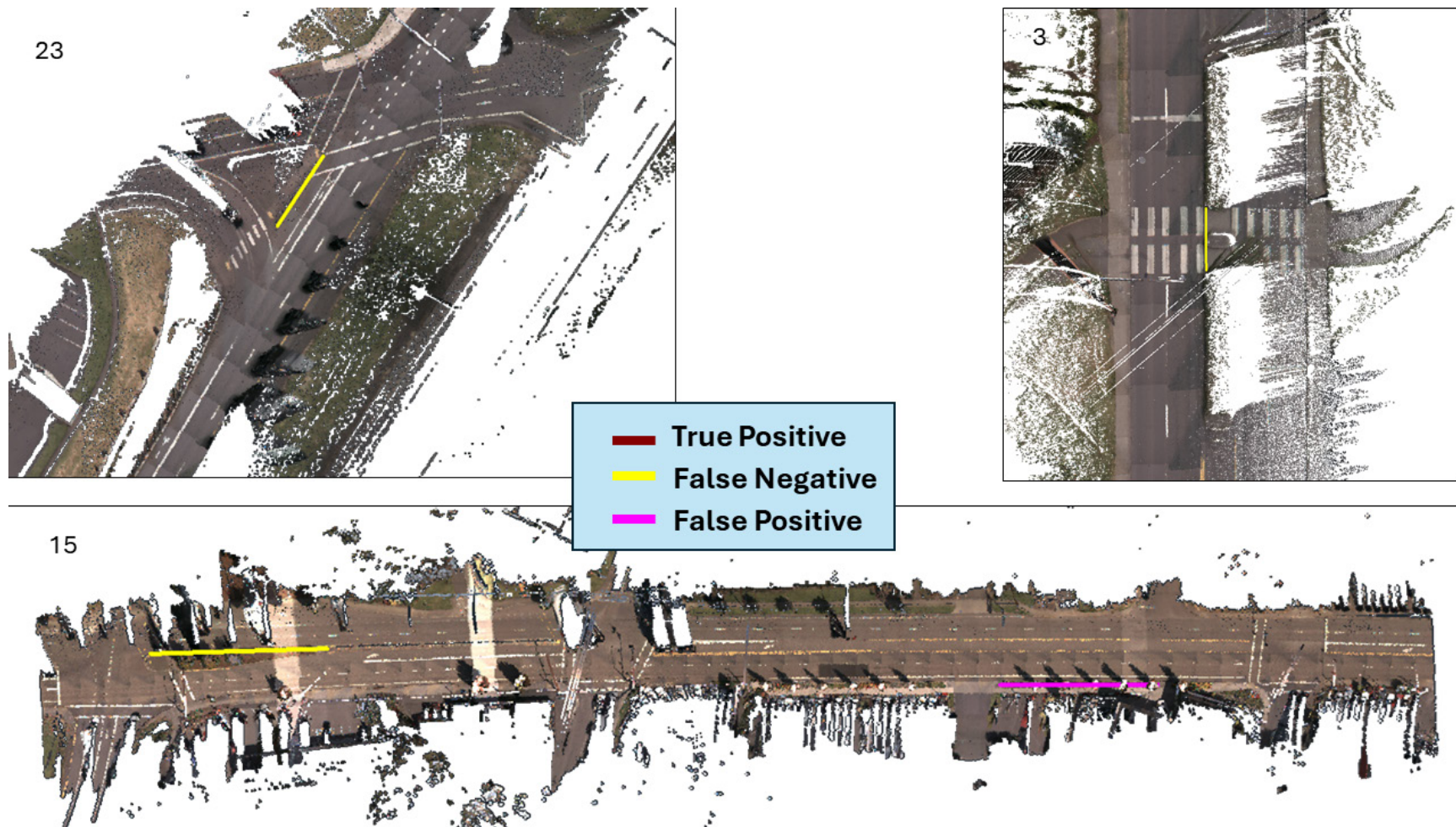


Figure 7-12. Median extraction results for the evaluation sections selected from Datasets 23, 3, and 15 showing True positive, false negatives, and false positives.

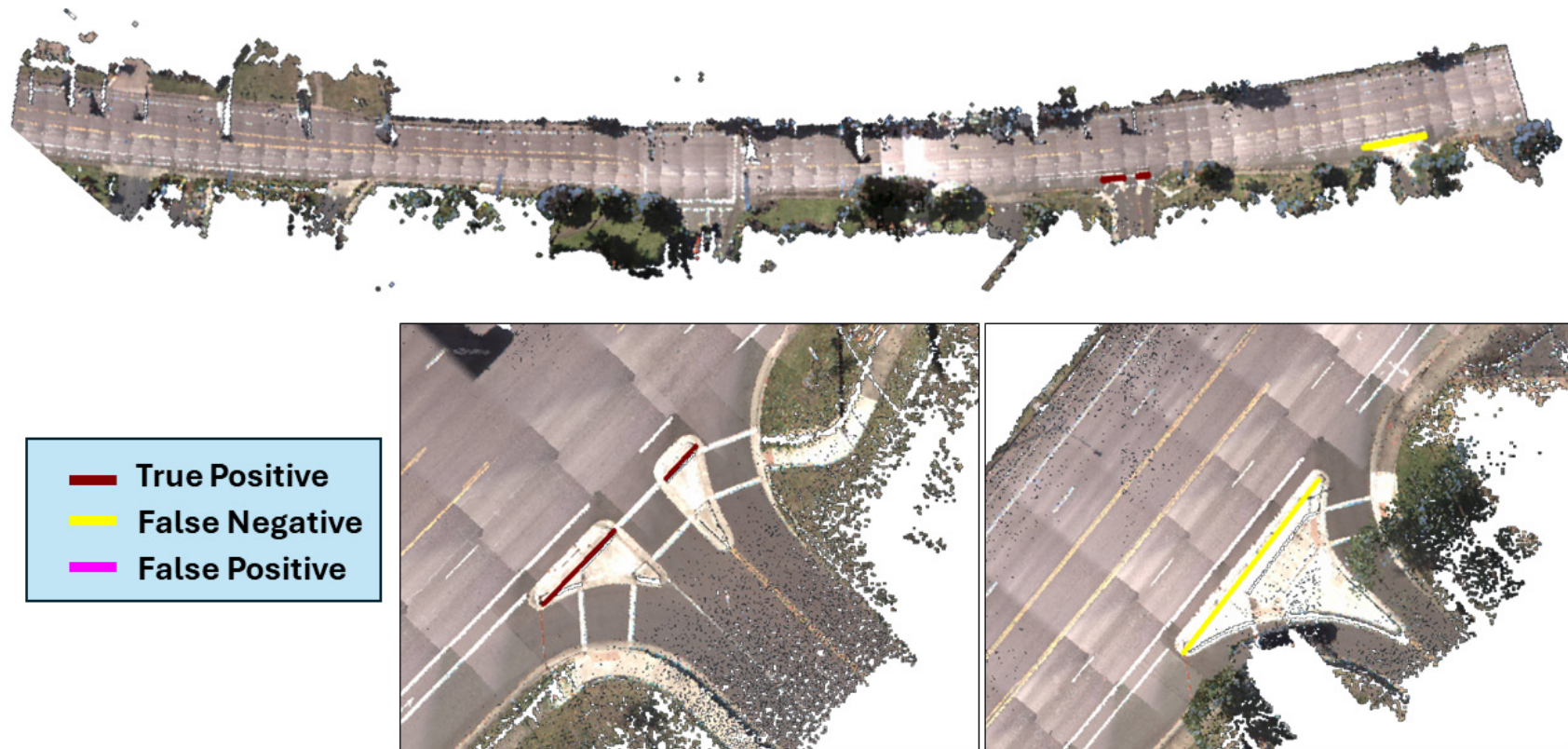


Figure 7-13. Median extraction results for the evaluation section selected from Dataset 32 showing true positives, false negatives, and false positives.

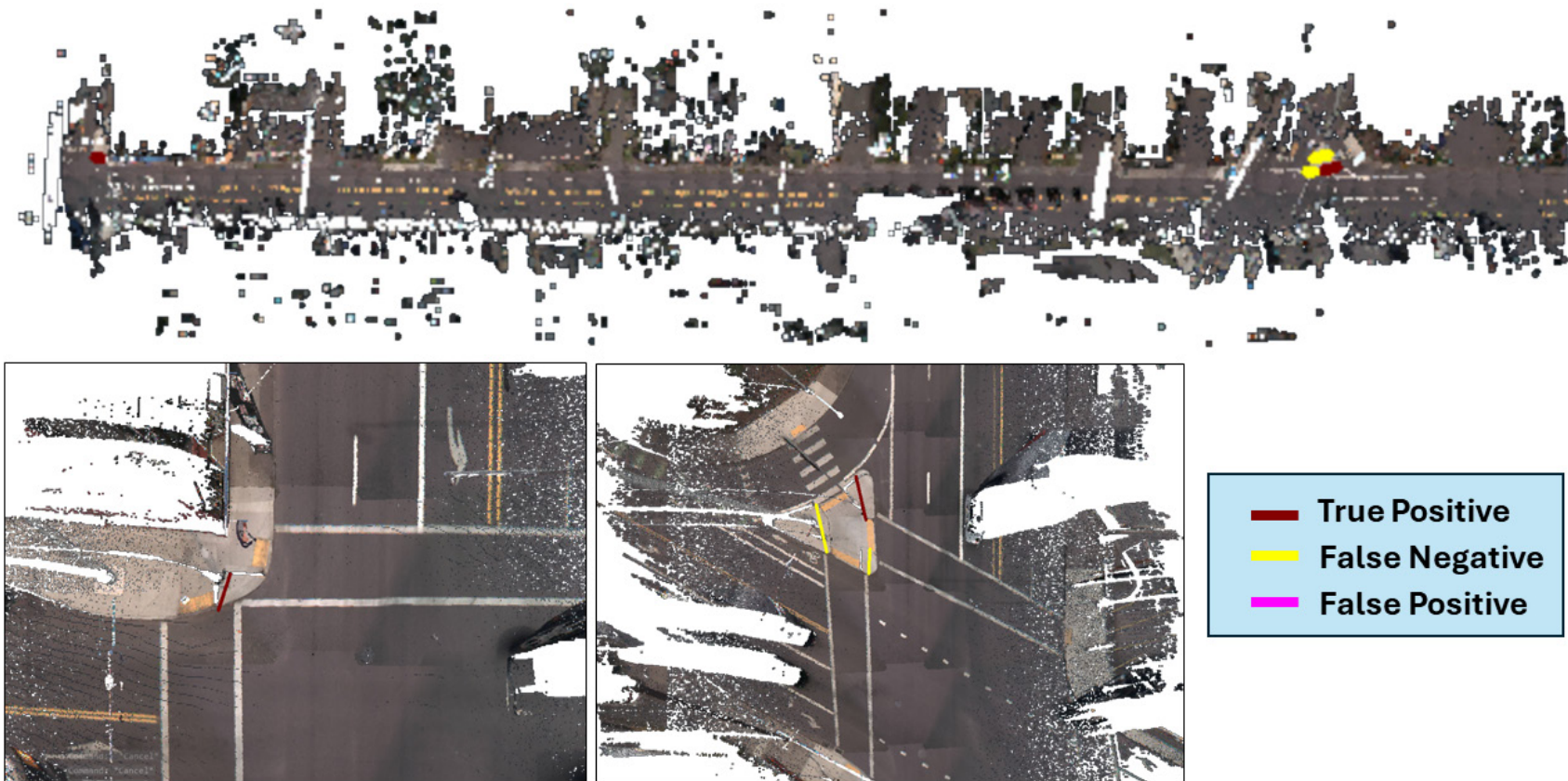


Figure 7-14. Median extraction results for the evaluation section selected from Dataset 2 showing true positives, false negatives, and false positives. In this case, a portion of the median was still captured.

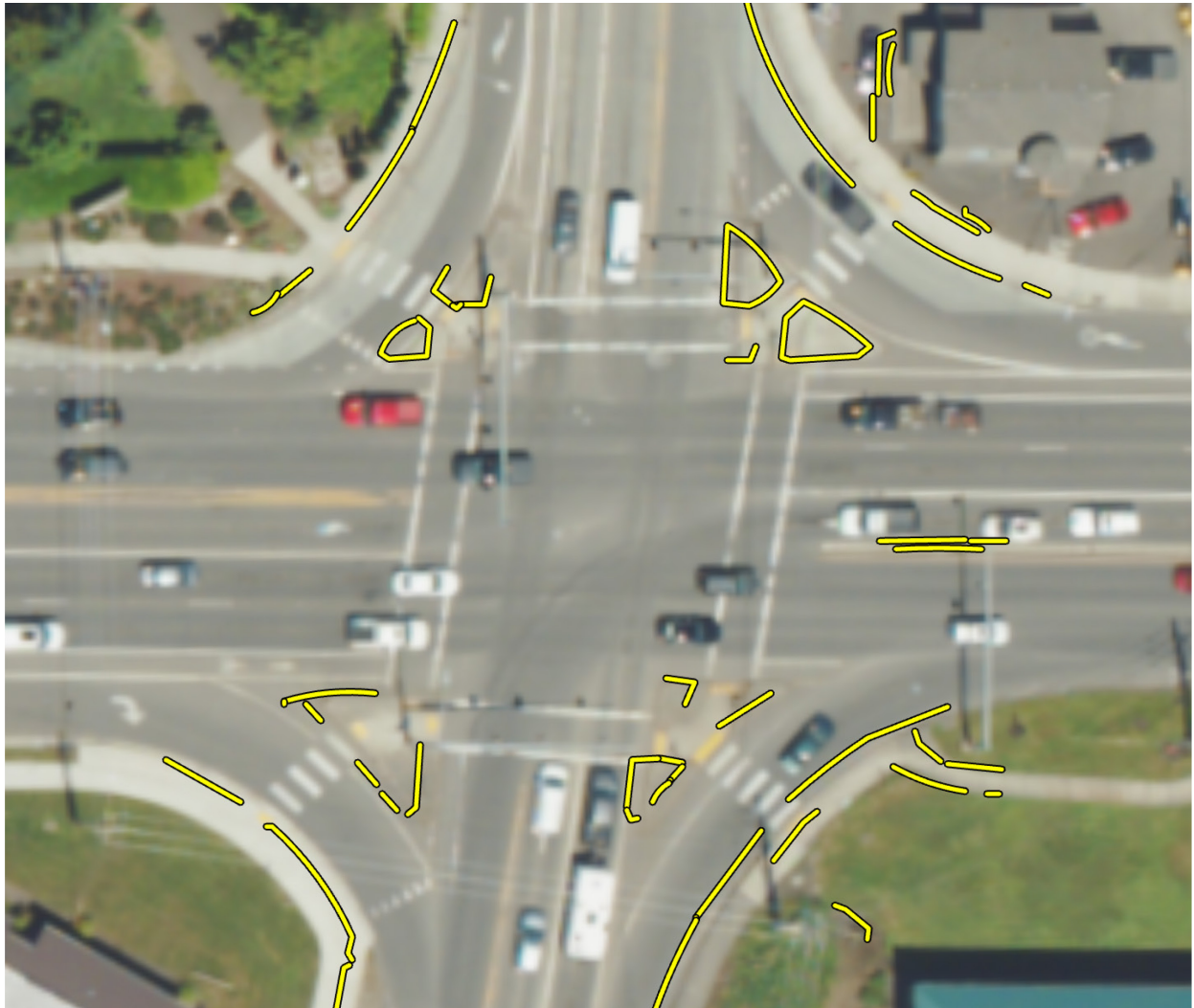


Figure 7-15. Example output of the curb extraction where medians can be clearly identified from the curb extraction.

7.3.4 Road marking width evaluation

ROME provides an estimate of the width of road markings. In order to evaluate the accuracy of the width estimate, sample measurements in CloudCompare were taken from several cross walk features in multiple datasets. Note that when there was substantial variability in the marking due to wear, multiple measurements were taken for that marking to obtain a representative value. The results are summarized separately for standard parallel and staggered continental crosswalks. ROME tends to underestimate the marking width by 30-40%. Much of this is likely the result of the rasterization process used by ROME. For these datasets a cell size of 0.1 m was used in the processing, which is similar to the average difference between the manual measurements and the ROME estimates.

Table 7-5. Summary of marking width accuracy assessment.

Metric	Standard Parallel		Staggered Continental	
	Difference (m)	Percent Difference	Difference (m)	Percent Difference
Average	0.073	26%	0.172	38%
Standard Deviation	0.029	8%	0.087	12%
Min	0.014	8%	0.030	13%
Max	0.138	41%	0.255	45%
RMS	0.078	27%	0.189	39%
Count	20	20	6	6

7.4 PROCESSING TIME EVALUATION

The time required for generating the data products were evaluated during the accuracy assessment. Table 7-6 summarizes both the manual and automatic processing time required to execute the SPR850 tools on the datasets. Note that these reported times assume each dataset to be processed individually. When processed in batch mode, the setup time would be substantially reduced as it only needs to be performed once for all of the datasets. Table 7-7 provides the time required for the manual digitization of those sections using Civil 3D.

Table 7-6. Summary of processing time required for executing the SPR850 tools.

Site	Dataset	Length (m)	# Points (millions)	Curb Extraction Setup Time (min)	Curb Extract Runtime (min)	Curb Extract GIS (min)	ROME tool setup (min)	ROME tool runtime (min)	Total (min)	Total (km/min)	Total (million points/min)
1	13	650	18	1	11	1	2	1	16	0.041	1.127
2	23	365	17	1	8	1	2	1	13	0.028	0.765
3	15	430	19	1	9	1	2	1	14	0.031	0.737
4	32	515	18	1	10	1	2	1	15	0.034	0.833
5	3	530	19	1	9	1	2	1	14	0.038	0.748
1	7	500	19	1	10	1	2	1	15	0.033	0.789
2	21	460	18	1	9	1	2	1	14	0.033	0.778
3	9	335	18	1	7	1	2	1	12	0.028	0.667
Overall		4785	166	10	88	10	18	9	155	0.031	0.935

Table 7-7. Comparison of processing time required for manual extraction in Civil3D.

Site	Dataset #	Section Length (m)	# Points (Millions)	Manual Digitization			
				Bike lanes (min)	Crosswalk (min)	Medians (min)	Total (min)
1	13	650	18	7	5	2	14
2	23	365	17	5	4	2	11
3	15	430	19	6	5	3	14
4	32	515	18	7	3	2	12
5	2	530	19	2	4	2	8
5	3	530	19	2	5	2	9
Overall		3020	110	29	26	13	68

7.4.1 Time savings

Note that the time shown in Table 7-6 include automated processing time which does not need user intervention during the process. In order to estimate the time savings through the SPR850 tools, the following considerations should be noted to put the analysis in context:

- Data are processed in batch mode so setup time is minimized and the analyst is working on other tasks while the automated portions of the workflow are underway. Additional efficiencies can be gained when processing larger sections of data in batch mode.
- The time to organize data is similar with both workflows. This is a conservative assumption since the manual procedure would typically require more time for the analyst to manage files; whereas the automated workflow has file organization built in. Additionally, the analyst will need to spend additional time during data import and export, which can quickly add up for large number of lidar files in practice.
- Ordinary, manual digitization will still require some cleanup and review for quality control. Such time was assumed to be similar between both the manual and SPR850 workflow.
- The SPR850 also provides detailed road marking, retroreflectivity analysis, and curb/median extraction, which were not digitized in the manual workflow. Hence, these added feature layers provide additional time savings beyond those quantified. Further time savings are available by expanding this automated workflow to include other assets or features of interest.
- The performance will vary based on the system utilized for processing. In this study computers with identical specifications and environment were utilized for both workflows.
- Manual cleanup time was estimated based on the accuracy of the results and considering the benefits provided by the input road marking and curb layers to help with the

digitization, re-labeling and other corrections. These were conservatively estimated based on the accuracy results presented in Section 7.3.

- The analyst is trained and proficient in executing both workflows.

Hence, the estimated time savings is a conservative estimate. From Table 7-8, the SPR850 workflow resulted in a 45%-time savings compared with the manual extraction workflow.

Table 7-8. Time savings computations based on the processing times in Table 7-6 and Table 7-7.

Procedure	Manual Extraction Workflow	SPR850 Manual Only	
	min/km	Multiplier	min/km
Curb Extraction Software	0.00		2.00
Curb GIS Operations	0.00		2.09
ROME	0.00		2.00
Bike lanes Digitization/Cleanup	9.60	0.20	1.92
Crosswalks Digitization/Cleanup	8.61	0.25	2.15
Medians Digitization/Cleanup	4.30	0.50	2.15
Total	22.52		12.32

7.4.2 Lane marking time savings

While the prior section assesses time savings for extracting bike lanes, crosswalks, and medians, this section focuses on the lane marking extraction portion of the workflow. Recently, Markovic et al. (2024) performed an evaluation of two semi-automated software tools (TopoDot and TerraScan) compared with manual extraction for a 500 m section. The key findings of that study are summarized as follows:

1. In TerraScan, an auxiliary polygon must be drawn to isolate the area of where the markings are located. The analyst must specify an intensity range from a histogram and also input the typical width of the markings. TopoDOT operates line by line. The analyst must draw a reference line as the seed to feed the automated process while providing the direction of the marking.
2. Overall good agreement (95% overlap) was observed with the road markings extracted from the tools with the manual results. In TopoDOT, the marking lines were slightly shorter (0.05 m) due to the difficulty in identifying the boundary of the lane marking with the intensity values.
3. Both software packages showed challenges in extracting complex markings such as painted medians or cross walks. The authors suggested that the intensity thresholds and average marking width parameter in TerraScan can be challenging for wider markings.

4. Both software packages struggled with worn markings due to insufficient contrast in intensity values between pavement and markings. These are also easy to miss in the manual extraction process.

The lane marking extraction time estimates from Markovic et al. (2024) are compared with ROME. Marcovic et al. (2024) evaluated a 500 m section, so those estimates were scaled to 1 km for efficiency. For the ROME computations, the time required for curb extraction was not included as that is not required for the lane marking extraction. It is assumed that the cleanup time would be similar between the three automated approaches. It is also worth noting that ROME also has a function of retroreflectivity assessment. Overall, from the summary results in Table 7-9, ROME was significantly faster (>80% time savings) than the two semi-automated software and the manual digitization procedures presented in Marcovic et al. (2024). Additionally, ROME is able to extract curved markings, which could require more time with the manual and semi-automated tools.

Table 7-9. Lane marking time savings from ROME compared with other approaches. Values from Markovic et al. (2024) were scaled to a 1 km section.

Methodology	Time (Min)	ROME Time savings (%)
ROME (without manual correction) (computed from Table 7-6)	3	N/A
TerraScan software (without manual correction) (Markovic 2024)	20	85%
TopoDOT software (without manual correction) (Markovic 2024)	16	81%
Manual Digitization (Markovic 2024)	60	95%

8.0 IMPLEMENTATION

The research team worked with representatives from the GIS, Geometronics, Engineering Automation, and Traffic Safety to develop this implementation plan for transitioning the research product(s) to operational use and identifying respective roles to facilitate this transition. Implementation work occurred throughout the project including meetings with ODOT and other agencies to discuss feature extraction needs.

During the course of the project, the research team developed many innovative algorithms and tested these on a variety of datasets with different conditions to provide a robust tool within the confines of the budget and timeline allocated for the project. However, there will likely be certain situations and datasets that the tools simply are not capable of processing. While much of the feature extraction work is automated through these tools are automated, there will need to be quality control and some manual cleanup in order to provide the final data products. Additionally, if there are objects that are not captured well in the mobile lidar data due to obstructions, those features cannot be extracted. The intent of the research was to keep manual cleanup work to a minimum.

For the tool to have optimal value, it will need continued usage and investment by ODOT. The following are recommended for effective implementation of this research products:

- Aggressively pursue funding opportunities for expanding use of innovative technologies within ODOT for asset management, safety, maintenance, operations, and other activities. Notably, the Advanced Digital Construction Management Systems (ADCMS) grant program in the Bipartisan Infrastructure Law (BIL) <https://www.fhwa.dot.gov/bipartisan-infrastructure-law/> has allocated \$100 million in grant funding over 5 years, building upon efforts such as the Every Day Counts program by the FHWA to encourage the adoption and implementation of technology.
- Continued training of staff to run the tools, perform quality control, and manual cleanup.
- Develop quality control procedures as part of the feature extraction workflows. For example, it is recommended to compare the extracted features directly with the point cloud as well as with other geospatial data sources such as aerial imagery to verify the accuracy and completeness of the extractions.
- Continued testing of the tools developed in the research to identify optimal parameters and at scale on pilot projects to showcase the tools. While a variety of scene types were considered for the research in order to produce robust tools, additional testing on other datasets will help continue to identify and resolve potential issues. In particular, the tools will need to be tested with the data from ODOT's new Riegl mobile laser scanning system, which was acquired near the end of this research project.
- Expand capabilities of the tools developed in the research, including:

- Explore usage in context of both (1) creating an initial inventory and (2) updating or verifying existing inventories to detect missing assets or assets that need attention.
- Develop new GIS-based tools to resolve common issues, clean up data files, and extract additional attributes by correlation with other data layers.
- Develop a methodology to integrate multiple passes of mobile data (e.g., northbound and southbound lanes) or extraction results from multiple years of data collections in order to produce a more complete inventory to avoid issues from occlusions or wet surfaces resulting in data loss on road markings or other features necessary to extract the pedestrian facilities.
- Invest in additional research projects to mine information from lidar and other geospatial data sources. Currently SPR866 is underway, which will add additional capabilities including extracting road geometric information (e.g., curvature, grade, cross slope) and pole-like objects.
- Develop a long-term maintenance plan for the tools. Invest in maintenance to provide updates to the software to ensure functionality and improvements to efficiency.
- Track basic efficiency measures to evaluate the return on investment provided by the tool and mobile lidar. Previous investments such as SPR799, which produced the RoME tool for road marking extraction only, have shown a high return on investment. A recent project using the tool to extract paint lines to determine paint quantities needed for restriping saved ODOT more money than the research initially cost. This project was awarded an AASHTO High Value Research award in 2023. It is likely that the tools developed in SPR850 will yield many benefits as well.

8.1 TOOL TESTING

Near the conclusion of the project, the project team worked with personnel from ODOT to test and evaluate the tools developed in this research. The testing plan and rubric provided for feedback is included in Appendix D. The testing had the following objectives:

- Inform ODOT staff about the tools and their capabilities.
- Ensure the tools execute properly on ODOT computers.
- Provide training to ODOT staff on executing the tools and working with the outputs.
- Provide feedback to the OSU research team on the tools and the manual.

Table 8-1 summarizes the four main tasks for the training portion of the research.

Table 8-1. Tasks associated with training for research products.

Training Task	Responsible Party	Timeframe	Description
1. Introduction and Tool Training	OSU	November 2023 (Milepost indexing tool) March 13, 2024 (Curb Extraction and ROME)	Demo with ODOT staff showing basic functionality and tool use and assessment rubric. Explained inputs and outputs of software.
2. Tool Testing and Performance Assessment	ODOT Staff	March-April 2024	ODOT staff deployed tool on small use case and record performance on rubric (Appendix E).
3. Presentation at NWTC	OSU	March 6, 2024	Summary presentations of SPR850, SPR866, and other related lidar feature extraction work
4. Tool Performance Reporting	All	April 2024	Research Team and ODOT staff met to discuss test result and plans for deployment.
5. Final TAC Meeting	All	April 4, 2024	Presentation from research team to TAC.

8.2 SOFTWARE USAGE

Detailed user manuals are provided with the tools to help users setup and become familiar with the software as well as adjust parameters when needed (Appendices A to C). The following should be considered for software installation:

- Software will need to be reviewed by ODOT i-Care prior to use by ODOT staff.
- All software consists of standalone executables and do not require installation.
- The software leverages intellectual property from OSU and EZDataMD LLC. The tools should not be shared outside of ODOT.
- For the SPR850 tool, installing the NVIDIA CUDA toolkit 11.2 can improve processing speeds. It is not required. <https://developer.nvidia.com/cuda-11.2.0-download-archive>
- For the EZFeat Curb Extraction and EZ Vox Bin N Grid tools, the Microsoft Visual Studio Redistributable Package is required. It is available at: <https://learn.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-170>. Note that it likely is already installed on the system from other software.
- For the ArcPy scripts, ArcGIS Pro software is required with a spatial analyst extension license.

9.0 CONCLUSIONS

This research explored the potential of extracting pedestrian and bicycle information from mobile lidar data- namely crosswalks, bike lanes, and median facilities. Several novel algorithms were developed in the course of the research to facilitate extraction of these features. First, leveraging prior work by the research team in ground filtering, road extraction, curb extraction and road marking extraction, several new tools were developed to execute these algorithms for automated extraction. These include (1) a batch script to efficiently execute several processes to extract curbs followed by a ArcPy script to generate GIS polyline features, and (2) a new version of the ROME tool which integrates the deep learning anssd rules-based algorithms described in Section 7.0 to extract road markings, cross walks, bike lanes and medians. Additionally, an extensive training database of annotated point clouds was developed.

Several important enhancements were made to the road marking extraction process, including a novel approach to cluster individual segments that are part of the same road marking in order to improve the classification of the longitudinal markings, cross walks, bike symbols, and other features. The markings serve as critical features to interpret the scene to detect cross walks, bike lanes, and some types of medians.

In general, the algorithm showed good performance and efficiency when tested on a variety of datasets. Features are extracted to a higher level of detail compared to current TransGIS data layers. (For example, rather than digitizing an entire crosswalk as a single polyline, ROME extracts each individual marking associated with the crosswalk). The SPR850 workflow (including cleanup) is estimated to save 45% of the time required for manual extraction of bike lanes, crosswalks, and medians). For road markings, ROME showed more than 80% time savings compared estimates for current extraction software tools and 95% savings over manual extraction. While some manual cleanup is required after implementing the tool, this work is kept to a minimum and is significantly aided by the output layers which help provide context when performing the extraction.

The products of this research ties directly into ODOT's mission by providing the capability to develop databases with detailed information on important traffic control devices that are critical to the safety of bicyclists and pedestrians in a multi-modal transportation environment and improving safety of ODOT personnel by minimizing the need to acquire this information in the field. These databases provide important information to guide decision making, planning, and maintenance efforts to improve the safety and mobility of the transportation network. By leveraging the existing data collected on a regular basis with the mobile lidar system, this research can reduce the significant time and cost required for collecting data about traffic safety devices. Further, the workflows to efficiently extract this information substantially reduce manual processing, resulting in a higher ROI on ODOT's mobile lidar system. Lastly, the framework integrating the tools developed in this research and from prior research can be extended to enable additional algorithms to be created with significantly reduced effort to extract additional features of interest such as jersey barriers, poles, parking stalls, curb ramps, and more. Some of this work is underway with SPR866, which focuses on computing road geometric properties such as grade, cross slope, and curvature from mobile lidar as well as a generalized framework for feature extraction from mobile lidar.

10.0 REFERENCES

- Ai, C., and Hou, Q. (2020). "Improving Pedestrian Infrastructure Inventory in Massachusetts Using Mobile LiDAR and Deep Learning." Nvidia GPU Technology Conference, March 23-26, Silicon Valley, CA. (Virtual event due to COVID-19).
- Ai, C., and Hou, Q. (2019). "Improving Pedestrian Infrastructure Inventory in Massachusetts Using Mobile LiDAR." Massachusetts Department of Transportation, Report No. 19-007.
- Ai, C. and Tsai, Y. (2016). "An Automated Sidewalk Assessment Method for the Americans with Disabilities Act Compliance Using 3-D Mobile LiDAR." Journal of Transportation Research Record Vol. 2542, 25-32.
- Ai, C. and Tsai, Y. (2016). "Automated Traffic Sign Retroreflectivity Condition Assessment Using Mobile LiDAR and Computer Vision." Transportation Research Part C - Emerging Technologies, 63, 96-113.
- Ai, C. and Tsai, Y. (2015). "Critical Assessment of an Enhanced Traffic Sign Detection Method Using Mobile LiDAR and INS Technologies." Journal of Transportation Engineering, 141(5), 04014096.
- Ai, C. and Tsai, Y. (2015). "Automatic Horizontal Curve Identification and Measurement Method Using GPS Data." Journal of Transportation Engineering, 141(2), 04014078.
- Azimi, S. M., Fischer, P., Körner, M., & Reinartz, P. (2018). Aerial LaneNet: Lane-marking semantic segmentation in aerial imagery using wavelet-enhanced cost-sensitive symmetric fully convolutional neural networks. IEEE Transactions on Geoscience and Remote Sensing, 57(5), 2920-2938.
- Baumgartner, A., Steger, C., Mayer, H., Eckstein, W., & Ebner, H. (1999). Automatic road extraction based on multi-scale, grouping, and context. Photogrammetric Engineering and Remote Sensing, 65, 777-786.
- Benjamin, S., & Gaydos, L. (1990). Spatial resolution requirements for automated cartographic road extraction. Photogrammetric Engineering and Remote Sensing, 56(1), 93-100.
- Boudet, L., & Midenet, S. (2009). Pedestrian crossing detection based on evidential fusion of video-sensors. Transportation research part C: emerging technologies, 17(5), 484-497.
- Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.
- Buslaev, A., Seferbekov, S., Iglovikov, V., & Shvets, A. (2018). Fully convolutional network for automatic road extraction from satellite imagery. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 207-210).

- Cabo, C., Kukko, A., García-Cortés, S., Kaartinen, H., Hyypä, J., & Ordoñez, C. (2016). An algorithm for automatic road asphalt edge delineation from mobile laser scanner data using the line clouds concept. *Remote Sensing*, 8(9), 740.
- Che, E., & Olsen, M. J. (2017). Fast ground filtering for TLS data via Scanline Density Analysis. *ISPRS Journal of Photogrammetry and Remote Sensing*, 129, 226-240.
- Che, E., & Olsen, M. J. (2017). Fast edge detection and segmentation of terrestrial laser scans through normal variation analysis. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4.
- Che, E., & Olsen, M. J. (2018). Multi-scan segmentation of terrestrial laser scanning data based on normal variation analysis. *ISPRS journal of photogrammetry and remote sensing*, 143, 233-248.
- Che, E., Jung, J., & Olsen, M. J. (2019). Object recognition, segmentation, and classification of mobile laser scanning point clouds: A state of the art review. *Sensors*, 19(4), 810.
- Che, E., Olsen, M.J., Parrish, C.E., and Jung, J. (2019b). "Pavement Marking Retroreflectivity Estimation and Evaluation using Mobile Lidar Data." *Photogrammetric Engineering and Remote Sensing*, 85(8), 573-583. ASPRS. <https://doi.org/10.14358/PERS.85.8.573>
- Che, E., & Olsen, M. J. (2019). An efficient framework for mobile lidar trajectory reconstruction and Mo-norvana segmentation. *Remote Sensing*, 11(7), 836.
- Che, E., Olsen, M. J., & Jung, J. (2021a). Efficient segment-based ground filtering and adaptive road detection from mobile light detection and ranging (LiDAR) data. *International Journal of Remote Sensing*, 42(10), 3633-3659.
- Che, E., Senogles, A., & Olsen, M. J. (2021b). *Vo-SmoG*: A Versatile, Smooth Segment-based Ground Filter for Point Clouds via Multi-scale Voxelization. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 8, 59-66.
- Chen, Z., Gao, B., & Devereux, B. (2017). State-of-the-art: DTM generation using airborne LIDAR data. *Sensors*, 17(1), 150.
- Cheng, M., Zhang, H., Wang, C., & Li, J. (2016). Extraction and classification of road markings using mobile laser scanning point clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(3), 1182-1196.
- Cheng, Y. T., Patel, A., Bullock, D., & Habib, A. (2020). LiDAR-based Lane Marking Extraction through Intensity Thresholding and Deep Learning Approaches: A Pavement-based Assessment. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43, 507-514.

- Dal Poz, A. P., & Yano Ywata, M. S. (2020). Adaptive random sample consensus approach for segmentation of building roof in airborne laser scanning point cloud. *International Journal of Remote Sensing*, 41(6), 2047-2061.
- Dong, Z., Yang, B., Hu, P., & Scherer, S. (2018). An efficient global energy optimization approach for robust 3D plane segmentation of point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 137, 112-133.
- Eltner, A., & Sofia, G. (2020). Structure from motion photogrammetric technique. In *Developments in Earth Surface Processes* (Vol. 23, pp. 1-24). Elsevier.
- El-Halawany, S., Moussa, A., Lichti, D. D., & El-Sheimy, N. (2011). Detection of road curb from mobile terrestrial laser scanner point cloud. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(5/W12).
- Famili, A., Sarasua, W. A., Shams, A., Davis, W. J., & Ogle, J. H. (2021). Application of Mobile Terrestrial LiDAR Scanning Systems for Identification of Potential Pavement Rutting Locations. *Transportation Research Record*, 03611981211005777.
- Fan, H., Yao, W., & Tang, L. (2013). Identifying man-made objects along urban road corridors from mobile LiDAR data. *IEEE Geoscience and Remote Sensing Letters*, 11(5), 950-954.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3), 273-297.
- Gruen, A., Agouris, P., & Li, H. (1995). Linear feature extraction with dynamic programming and globally enforced least squares matching. In *Automatic Extraction of Man-Made Objects from Aerial and Space Images* (pp. 83-94). Birkhäuser Basel.
- Guan, H., Li, J., Yu, Y., Wang, C., Chapman, M., & Yang, B. (2014). Using mobile laser scanning data for automated extraction of road markings. *ISPRS Journal of Photogrammetry and Remote Sensing*, 87, 93-107.
- Guan, H., Li, J., Yu, Y., Chapman, M., & Wang, C. (2014). Automated road information extraction from mobile laser scanning data. *IEEE Transactions on Intelligent Transportation Systems*, 16(1), 194-205.
- Guo, J., Tsai, M. J., & Han, J. Y. (2015). Automatic reconstruction of road surface features by using terrestrial mobile lidar. *Automation in Construction*, 58, 165-175.
- Hallmark, S.L., (2001). GPS to LRM: Integration of Spatial Point Features with Linear Referencing Model, US DOT Project MTC-A-4 and Iowa DOT Project 00-68. http://publications.iowa.gov/21323/1/IADOT_CTRE_00_68_GPS_LRM_Integration_Spatial_Point_Linear_Referencing_Methods_2001.PDF

- Hecht-Nielsen, R. (1992). Theory of the backpropagation neural network. In *Neural networks for perception* (pp. 65-93). Academic Press.
- Hernández, J., & Marcotegui, B. (2009, September). Filtering of artifacts and pavement segmentation from mobile lidar data. In *ISPRS Workshop Laserscanning 2009*.
- Huang, J., & You, S. (2016, December). Point cloud labeling using 3d convolutional neural network. In *2016 23rd International Conference on Pattern Recognition (ICPR)* (pp. 2670-2675). IEEE.
- Hoang, T. M., Nam, S. H., & Park, K. R. (2019). Enhanced detection and recognition of road markings based on adaptive region of interest and deep learning. *IEEE Access*, 7, 109817-109832.
- Husain, A., & Vaishya, R. C. (2016, October). A time efficient algorithm for ground point filtering from mobile LiDAR data. In *2016 International Conference on Control, Computing, Communication and Materials (ICCCCM)* (pp. 1-5). IEEE.
- Hou, Q., and Ai, C. (2020). "A Network-Level Sidewalk Inventory Method Using Mobile LiDAR and Deep Learning." *Transportation Research Part C: Emerging Technologies*, 119, 102772.
- Hu, W., Wang, W., Ai, C., Wang, J., Wang, W., Meng, X., Liu, J., Tao, H., and Qiu, S. (2021). "Machine Vision-based Surface Crack Analysis for Transportation Infrastructure." *Automation in Construction*, 132(2021), 103973.
- Ibrahim, S., & Lichti, D. (2012). Curb-based street floor extraction from mobile terrestrial LiDAR point cloud. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci*, 39, B5.
- IOWA DOT. (2019). Medians. <https://iowadot.gov/design/dmanual/03e-01.pdf>
- Ishino, Y., & Saji, H. (2008, August). Extraction of road markings from aerial images. In *2008 SICE Annual Conference* (pp. 2180-2183). IEEE.
- Jin, H., Feng, Y., & Li, Z. (2009). Extraction of road lanes from high-resolution stereo aerial imagery based on maximum likelihood segmentation and texture enhancement. In *2009 Digital Image Computing: Techniques and Applications* (pp. 271-276). IEEE.
- Jin, H., Miska, M., Chung, E., Li, M., & Feng, Y. (2012). Road feature extraction from high resolution aerial images upon rural regions based on multi-resolution image analysis and Gabor filters. *Remote Sensing-Advanced Techniques and Platforms*, 387-414.
- Jung, J., Che, E., Olsen, M. J., & Parrish, C. (2019). Efficient and robust lane marking extraction from mobile lidar point clouds. *ISPRS journal of photogrammetry and remote sensing*, 147, 1-18. <https://doi.org/10.1016/j.isprsjprs.2018.11.012>.

- Jung, J., M. Olsen, E. Che, and C. Parrish. 2020. "Efficient Extraction and Evaluation of Complex Pavement Markings from Mobile Laser Scan Data." Accessed September 10, 2023. <https://digital.lib.washington.edu/researchworks/handle/1773/46275>.
- Kasemsuppakorn, P., & Karimi, H. A. (2013). Pedestrian network extraction from fused aerial imagery (orthoimages) and laser imagery (lidar). *Photogrammetric Engineering & Remote Sensing*, 79(4), 369-379.
- Kim, J. G., Han, D. Y., Yu, K. Y., Kim, Y. I., & Rhee, S. M. (2006). Efficient extraction of road information for car navigation applications using road pavement markings obtained from aerial images. *Canadian Journal of Civil Engineering*, 33(10), 1320-1331.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 1097-1105.
- Kumar, P., McElhinney, C. P., Lewis, P., & McCarthy, T. (2013). An automated algorithm for extracting road edges from terrestrial mobile LiDAR data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 85, 44-55.
- Kumar, P., McElhinney, C. P., Lewis, P., & McCarthy, T. (2014). Automated road markings extraction from mobile laser scanning data. *International Journal of Applied Earth Observation and Geoinformation*, 32, 125-137.
- Kumar, P., Lewis, P., & McElhinney, C. P. (2018). Point Cloud Data Retrieval from 3D Geospatial Database for Automated Road Median Extraction.
- Kurz, F., Azimi, S. M., Sheu, C. Y., & d'Angelo, P. (2019). Deep learning segmentation and 3D reconstruction of road markings using multiview aerial imagery. *ISPRS International Journal of Geo-Information*, 8(1), 47.
- Lausser, L., Schwenker, F., & Palm, G. (2008, April). Detecting zebra crossings utilizing AdaBoost. In *ESANN* (pp. 535-540).
- Lawin, F. J., Danelljan, M., Tosteberg, P., Bhat, G., Khan, F. S., & Felsberg, M. (2017, August). Deep projective 3D semantic segmentation. In *International Conference on Computer Analysis of Images and Patterns* (pp. 95-107). Springer, Cham.
- Li, Y., R. Bu, M. Sun, W. Wu, X. Di, and B. Chen (2018). PointCNN: Convolution on X-Transformed Points, in *Proceedings of the Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018 (NeurIPS 2018)*, Montreal, Canada, 828.

- Liu, L., Ma, H., Chen, S., Tang, X., Xie, J., Huang, G., & Mo, F. (2020). Image-Translation-Based Road Marking Extraction from Mobile Laser Point Clouds. *IEEE Access*, 8, 64297-64309.
- Lin, X., & Zhang, J. (2015). Segmentation-based ground points detection from mobile laser scanning point cloud. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(7), 99.
- Lin, Y., Wang, C., Chen, B., Zai, D., & Li, J. (2017). Facet segmentation-based line segment extraction for large-scale point clouds. *IEEE Transactions on Geoscience and Remote Sensing*, 55(9), 4839-4854.
- Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440).
- Ma, L., Li, Y., Li, J., Yu, Y., Junior, J. M., Gonçalves, W. N., & Chapman, M. A. (2020). Capsule-based networks for road marking extraction and classification from mobile LiDAR point clouds. *IEEE Transactions on Intelligent Transportation Systems*.
- Maboudi, M., Amini, J., Malihi, S., & Hahn, M. (2018). Integrating fuzzy object based image analysis and ant colony optimization for road extraction from remotely sensed images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 138, 151-163.
- Maalek, R., Lichti, D. D., & Ruwanpura, J. Y. (2018). Robust segmentation of planar and linear features of terrestrial laser scanner point clouds acquired from construction sites. *Sensors*, 18(3), 819.
- Mahmoudabadi, H., Olsen, M. J., & Todorovic, S. (2016). Efficient terrestrial laser scan segmentation exploiting data structure. *ISPRS journal of photogrammetry and remote sensing*, 119, 135-150.
- Markovic, M., Manojlovic, M.D., Vujinovic, M., Batilovic, M., and Krnic, D. (2024). Advancements in road markings detection using mobile mapping scanner-derived point cloud. 9th International Conference “Civil Engineering-Science and Practice”. GNP 2024, Kolasin, Montenegro.
- Meng, X., Currit, N., & Zhao, K. (2010). Ground filtering algorithms for airborne LiDAR data: A review of critical issues. *Remote Sensing*, 2(3), 833-860.
- Mikhail, E. M., Bethel, J. S., & McGlone, J. C. (2001). *Introduction to modern photogrammetry*. New York, 19.

- Miraliakbari, A., Hahn, M., & Sok, S. (2015). Automatic extraction of road surface and curbstone edges from mobile laser scanning data. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(4), 119.
- Miyazaki, R., Yamamoto, M., Hanamoto, E., Izumi, H., & Harada, K. (2014). A LINE-BASED APPROACH FOR PRECISE EXTRACTION OF ROAD AND CURB REGION FROM MOBILE MAPPING DATA. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 2(5).
- NACTO, (2022). Urban Bikeway Design Guide. <https://nacto.org/publication/urban-bikeway-design-guide/bike-lanes/buffered-bike-lanes/>
- NITC, (2019). Biking Safely Through the Intersection: Guidance for Protected Bike Lanes. <https://nitc.trec.pdx.edu/news/biking-safely-through-intersection-guidance-protected-bike-lanes>
- Olsen, M.J., Roe, G.V., Glennie, C., Persi, F., Reedy, M., Hurwitz, D., Williams, K., Tuss, H., Squellati, A., & Knodler, M. (2013a). “Guidelines for the use of mobile lidar in transportation applications,” TRB NCHRP Final Report 748. <http://www.trb.org/Publications/Blurbs/169111.aspx> , <http://learnmobilelidar.com>
- Olsen, M. J., G. V. Roe, and J. Raugust. (2013b). Use of advanced geospatial data tools, technologies, and Information in DOT projects. NCHRP Synthesis 446, Topic 43-09. Washington, DC: National Academies of Sciences, Engineering, and Medicine.
- Olsen, M.J., Massey, C., Senogles, A., Leshchinsky, B.A., and Wartman, J. (2020). Predicting seismically induced rockfall hazard for targeted site mitigation, ODOT SPR809 Final Report.
- Olsen, M.J., Parrish, C.E., Che, E., Jung, J., and Greenwood, J. (2018). Lidar for maintenance of pavement reflective markings and retroreflective signs. Vol I. Reflective Markings, Oregon DOT SPR-799 Final Report. <https://www.oregon.gov/ODOT/Programs/ResearchDocuments/SPR799%20Lidar%20for%20Maintenance%20of%20Pavement%20Reflective%20Markings%20and%20Retroreflective%20Signs%20Vol.%20I%20Reflective%20Pavement%20Markings.pdf>
- Olsen, M.J., Parrish, C.E., Che, E., Jung, J., and Greenwood, J. (2018). Lidar for maintenance of pavement reflective markings and retroreflective signs. Vol II. Retroreflective Signs, Oregon DOT SPR-799 Final Report. <https://www.oregon.gov/ODOT/Programs/ResearchDocuments/SPR799LidarRetroFinalReport.pdf>
- Oregon DOT (2002). Desired Conditions of Maintenance Features on State Highways.

- Oregon DOT (2015b). Mobile Lidar. Retrieved from <https://www.oregon.gov/ODOT/ETA/Pages/LiDAR.aspx>
- ODOT (2020) Bike-Pedestrian Data Collection Guide. <https://www.oregon.gov/odot/Programs/AssetMgmt/BikePed-Data-Collection-Guide.pdf>
- ODOT (2022) Statewide Equity Layer. <https://arcg.is/fn4HP>
- ODOT. 2023. "Traffic line manual." Accessed December 26, 2023.
- Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1), 62-66.
- Patel, A., Cheng, Y. T., Ravi, R., Lin, Y. C., Bullock, D., & Habib, A. (2021). Transfer Learning for LiDAR-Based Lane Marking Detection and Intensity Profile Generation. *Geomatics*, 1(2), 287-309.
- Pigeon, L., Moulin, B., Solaiman, B., Toutin, T., & Thomson, K. P. (1999, June). Human-experts features extraction strategies for topographical map production and updating. In *IEEE 1999 International Geoscience and Remote Sensing Symposium. IGARSS'99* (Cat. No. 99CH36293) (Vol. 1, pp. 296-298). IEEE.
- Qi, C.R., H. Su, K. Mo, and L.J. Guibas, (2017). PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 652.
- Qi, C.R., L. Yi, H. Su, and L.J. Guibas (2017). PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, in *Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, Long Beach, California, 5099.
- Qiu, Z., Zhuang, Y., Yan, F., Hu, H., & Wang, W. (2018). RGB-DI images and full convolution neural network-based outdoor scene understanding for mobile robots. *IEEE Transactions on Instrumentation and Measurement*, 68(1), 27-37.
- Quackenbush, L. J. (2004). A review of techniques for extracting linear features from imagery. *Photogrammetric Engineering & Remote Sensing*, 70(12), 1383-1392.
- Ravi, R., Bullock, D., and Habib, A. (2021). "Pavement Distress and Debris Detection using a mobile mapping system with 2D profiler lidar, *Transportation Research Record* 2675(9), 428-438.
- Riveiro, B., González-Jorge, H., Martínez-Sánchez, J., Díaz-Vilariño, L., & Arias, P. (2015). Automatic detection of zebra crossings from mobile LiDAR data. *Optics & Laser Technology*, 70, 63-70.

- Rodriguez-Cuenca, B., Garcia-Cortes, S., Ordóñez, C., & Alonso, M. C. (2015). An approach to detect and delineate street curbs from MLS 3D point cloud data. *Automation in Construction*, 51, 103-112.
- Rodríguez-Cuenca, B., García-Cortés, S., Ordóñez, C., & Alonso, M. C. (2016). Morphological operations to extract urban curbs in 3D MLS point clouds. *ISPRS International Journal of Geo-Information*, 5(6), 93.
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer, Cham.
- Roynard, X., Deschaut, J. E., & Goulette, F. (2018). Classification of point cloud scenes with multiscale voxel deep network. *arXiv preprint arXiv:1804.03583*.
- Sakr, G., Berberian, A., & Habib, P. (2019). Comparing Deep Learning Models for Road Asset Detection and Classification in LiDAR Point Cloud. In *2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)* (pp. 138-145). IEEE.
- Serna, A., & Marcotegui, B. (2014). Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93, 243-255.
- Serna, A., & Marcotegui, B. (2013). Urban accessibility diagnosis from mobile laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 84, 23-32.
- Shams, A., Sarasua, W. A., Putman, B. J., Davis, W. J., & Ogle, J. H. (2020). Highway Cross-Sectional Design and Maintenance to Minimize Hydroplaning. *Journal of Transportation Engineering, Part B: Pavements*, 146(4), 04020065.
- Shams, A., Sarasua, W., Famili, A., Davis, W., Ogle, J. Cassule, L., and Mammadrahimli, A. (2018) "Highway Cross-Slope Measurement using Mobile LiDAR" *Transportation Research Record* 2672, no.39, pp 88-97.
- Sithole, G., & Vosselman, G. (2004). Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds. *ISPRS journal of photogrammetry and remote sensing*, 59(1-2), 85-101.
- Soilán, M., Riveiro, B., Martínez-Sánchez, J., & Arias, P. (2017). Segmentation and classification of road markings using MLS data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 123, 94-103.
- Song, M., & Civco, D. (2004). Road extraction using SVM and image segmentation. *Photogrammetric Engineering & Remote Sensing*, 70(12), 1365-1371.

- Stirling, D.M., 1990. Close-range photogrammetry, in *Engineering Surveying Technology* (T.J.M. Kennie and G. Petrie, Eds.) CRC Press, Taylor & Francis Group, Boca Raton, Florida
- Taniguchi, R. I., & Kawaguchi, E. (1989, July). Road network extraction from Landsat TM image. In *Third International Conference on Image Processing and its Applications*, 1989. (pp. 222-226). IET.
- Tao, C., Qi, J., Li, Y., Wang, H., & Li, H. (2019). Spatial information inference net: Road extraction using road-specific contextual information. *ISPRS Journal of Photogrammetry and Remote Sensing*, 158, 155-166.
- Teo, T. A., & Yu, H. L. (2015). Empirical radiometric normalization of road points from terrestrial mobile LiDAR system. *Remote Sensing*, 7(5), 6336-6357.
- Thompson, M. M., Eller, R. C., Radlinski, W. A., & Speert, J. L. (Eds.). (1966). *Manual of photogrammetry* (Vol. 1, p. 61). Falls Church, VA: American Society of Photogrammetry.
- Tonkin, T. N., Midgley, N. G., Graham, D. J., & Labadz, J. C. (2014). The potential of small unmanned aircraft systems and structure-from-motion for topographic surveys: A test of emerging integrated approaches at Cwm Idwal, North Wales. *Geomorphology*, 226, 35-43.
- Tsai, Y. and Ai, C.. (2017). “Automated Superelevation Measurement Method Using a Low-Cost Mobile Device.” *Journal of Transportation Research Record*, Vol. 2621, 62-70.
- Tsai, Y., Ai, C., Wang Z., and Pitts, E. (2013). “A Mobile Cross Slope Measurement Method Using LiDAR Technology.” *Journal of Transportation Research Record*, Vol. 2367, 53-62.
- Tsai, Y., Wu, Y.*, Ai, C. and Pitts, E. (2012). “Critical Assessment of Measuring Concrete Joint Faulting Using 3-D Continuous Pavement Profile Data.” *Journal of Transportation Engineering*, 138(11), 1291-1296.
- Vo, A. V., Truong-Hong, L., Laefer, D. F., & Bertolotto, M. (2015). Octree-based region growing for point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 104, 88-100.
- Wang, K., Wang, T., & Liu, X. (2019). A review: Individual tree species classification using integrated airborne LiDAR and optical imagery with a focus on the urban environment. *Forests*, 10(1), 1.
- Wang, L., Huang, Y., Shan, J., & He, L. (2018). MSNet: Multi-scale convolutional network for point cloud classification. *Remote Sensing*, 10(4), 612.

- Wang, Q., & Kim, M. K. (2019). Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018. *Advanced Engineering Informatics*, 39, 306-319.
- Wang, X., Chan, T. O., Liu, K., Pan, J., Luo, M., Li, W., & Wei, C. (2020). A robust segmentation framework for closely packed buildings from airborne LiDAR point clouds. *International Journal of Remote Sensing*, 41(14), 5147-5165.
- Wen, C., Sun, X., Li, J., Wang, C., Guo, Y., & Habib, A. (2019). A deep learning framework for road marking extraction, classification and completion from mobile laser scanning point clouds. *ISPRS journal of photogrammetry and remote sensing*, 147, 178-192.
- Weng, Q. (2012). Remote sensing of impervious surfaces in the urban areas: Requirements, methods, and trends. *Remote Sensing of Environment*, 117, 34-49.
- Westoby, M. J., Brasington, J., Glasser, N. F., Hambrey, M. J., & Reynolds, J. M. (2012). 'Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, 179, 300-314.
- Wolf, J., Pietz, T., Richter, R., Discher, S., & Döllner, J. (2021). Image-based Road Marking Classification and Vector Data Derivation from Mobile Mapping 3D Point Clouds. In *VISIGRAPP (4: VISAPP)* (pp. 227-234).
- Wolf, P., Dewitt, B., & Wilkinson, B. (2014). *Elements of Photogrammetry with Applications in GIS*. McGraw-Hill Education.
- Wu, B., Yu, B., Huang, C., Wu, Q., & Wu, J. (2016). Automated extraction of ground surface along urban roads from mobile laser scanning point clouds. *Remote Sensing Letters*, 7(2), 170-179.
- Wu, W., Z. Qi, and F. Li, (2019). PointConv: Deep Convolutional Networks on 3D Point Clouds, *Proceedings of the IEEE-CVF Confer-ence on Computer Vision and Pattern Recognition*, 9621.
- Xia, S., Chen, D., Wang, R., Li, J., & Zhang, X. (2020). Geometric primitives in LiDAR point clouds: A review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13, 685-707.
- Xu, S., Wang, R., & Zheng, H. (2016). Road curb extraction from mobile LiDAR point clouds. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2), 996-1009.
- Xu, Y., Xie, Z., Feng, Y., & Chen, Z. (2018). Road extraction from high-resolution remote sensing imagery using deep learning. *Remote Sensing*, 10(9), 1461.

- Xu, Y., X. Shen, and S. Lim, (2021) CorDet: Corner-Aware 3D Object Detection Networks for Automated Scan-to-BIM, *Journal of Computing in Civil Engineering* 35(3), 04021002.
- Yadav, M., Singh, A. K., & Lohani, B. (2017). Extraction of road surface from mobile LiDAR data of complex road environment. *International Journal of Remote Sensing*, 38(16), 4655-4682.
- Yang, B., Fang, L., Li, Q., & Li, J. (2012). Automated extraction of road markings from mobile LiDAR point clouds. *Photogrammetric Engineering & Remote Sensing*, 78(4), 331-338.
- Yang, B., Dong, Z., Zhao, G., & Dai, W. (2015). Hierarchical extraction of urban objects from mobile laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 99, 45-57.
- Yang, J., Kang, Z., Cheng, S., Yang, Z., & Akwensi, P. H. (2020). An individual tree segmentation method based on watershed algorithm and three-dimensional spatial distribution analysis from airborne LiDAR point clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13, 1055-1067.
- Yang, M., Wan, Y., Liu, X., Xu, J., Wei, Z., Chen, M., & Sheng, P. (2018). Laser data based automatic recognition and maintenance of road markings from MLS system. *Optics & Laser Technology*, 107, 192-203.
- Yan, L., Liu, H., Tan, J., Li, Z., Xie, H., & Chen, C. (2016). Scan line based road marking extraction from mobile LiDAR point clouds. *Sensors*, 16(6), 903.
- Yao, L., Qin, C., Chen, Q., & Wu, H. (2021). Automatic road marking extraction and vectorization from vehicle-borne laser scanning data. *Remote Sensing*, 13(13), 2612.
- Yu, Y., Li, J., Guan, H., Jia, F., & Wang, C. (2014). Learning hierarchical features for automated extraction of road markings from 3-D mobile LiDAR point clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(2), 709-726.
- Yu, Y., Li, Y., Liu, C., Wang, J., Yu, C., Jiang, X., ... & Zhang, Y. (2021). MarkCapsNet: Road Marking Extraction From Aerial Images Using Self-Attention-Guided Capsule Network. *IEEE Geoscience and Remote Sensing Letters*, 19, 1-5.
- Zai, D., Li, J., Guo, Y., Cheng, M., Lin, Y., Luo, H., & Wang, C. (2017). 3-D road boundary extraction from mobile laser scanning data via supervoxels and graph cuts. *IEEE Transactions on Intelligent Transportation Systems*, 19(3), 802-813.
- Zaboli, M., Rastiveis, H., Shams, A., Hosseiny, B., and Sarasua, W.A. (2019) "Classification of Mobile Terrestrial LiDAR Point Clouds in Urban Area Using Local Features" *ISPRS International Joint Conferences SMPR and GIREsearch. The Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, Vol: XLII-4/W18, pp 1117-1122.

- Zhang, Y., Yang, W., Liu, X., Wan, Y., Zhu, X., & Tan, Y. (2021). Unsupervised Building Instance Segmentation of Airborne LiDAR Point Clouds for Parallel Reconstruction Analysis. *Remote Sensing*, 13(6), 1136.
- Zhao, C., Guo, H., Lu, J., Yu, D., Zhou, X., & Lin, Y. (2021). A new approach for roof segmentation from airborne LiDAR point clouds. *Remote Sensing Letters*, 12(4), 377-386.
- Zhong, L., Liu, P., Wang, L., Wei, Z., Guan, H., & Yu, Y. (2016). A Combination of Stop-and-Go and Electro-Tricycle Laser Scanning Systems for Rural Cadastral Surveys. *ISPRS International Journal of Geo-Information*, 5(9), 16
- Zhou, Y., & Tuzel, O. (2018). Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4490-4499).
- Zhu, X., Liu, X., Zhang, Y., Wan, Y., & Duan, Y. (2021). Robust 3-D plane segmentation from airborne point clouds based on quasi-a-contrario theory. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, 7133-7147.
- Zhuang, Y., He, G., Hu, H., & Wu, Z. (2015). A novel outdoor scene-understanding framework for unmanned ground vehicles with 3D laser scanners. *Transactions of the Institute of Measurement and Control*, 37(4), 435-445.

APPENDIX: A

APPENDIX A: MILEPOST INDEXING TOOL

A.1 INTRODUCTION

The milepost indexing tool links the mobile lidar trajectory file(s) as kmz(s) with a GIS mile post layer to map the milepost values to the mobile lidar trajectory file. This can then be linked to identify las/laz files for sections of interest corresponding to a specific milepost range and associated highway. This tool is provided as a zip file with the following components:

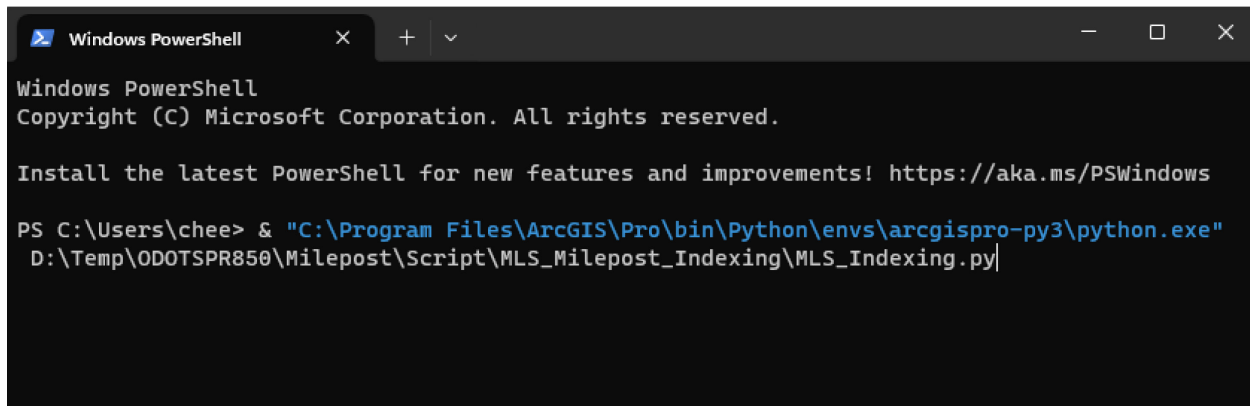
- *Idx*: This folder contains an example output index file that can be queried to identify which mobile lidar files correspond to specific milepost ranges on specific highways.
- *ODOT_MPtenths.gdb* – contains the geodatabase with a point feature class with mileposts at 1/10 mile spacing.
- *Script* – contains the arcpy script to link the kmz files with the milepost feature class.
- *Trj* – contains kmz files with the trajectory information for different mobile lidar datasets.

A.2 RUNNING THE SCRIPT

The python script "MLS_Indexing.py" is located in the Script/MLS_Milepost_Indexing folder. Before running the script, it needs to be configured through the following process:

- Open the script with a text editor, change the input argument near the top of the script. *Be sure to use "/" or "\" in the path.*
 - *dir_trj*: the path to the directory containing all of the trajectory kml files. The script will go through all the sub-folders to find all the kml files.
 - *gdb_mp*: the geodatabase files storing the milepost geodatabase.
 - *range_align*: search range for path alignment. (default: 30 m, the unit should align with the unit of the milepost geodatabase)
 - *range_join*: search range for joining the information from the milepost. (default: 100 m, the unit should align with the unit of the milepost geodatabase)
 - *dir_idx*: directory that the output csv file will be stored in. ***This folder needs to exist.***
- Save the script.
- Open a Windows Powershell and run the script (Figure X-X) by:
 - Input "&" followed by a space.
 - Drag and drop the python.exe in the ArcGIS Pro to the command prompt. Typically, it will be "c:/Program Files/ArcGIS/Pro/bin/Python/envs/arcgispro-py3/python.exe"

- Input a space.
- Drag and drop the python script to the command prompt.
- Press enter to run the script.



```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\chee> & "C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3\python.exe"
D:\Temp\ODOTSPR850\Milepost\Script\MLS_Milepost_Indexing\MLS_Indexing.py

```

Figure A-0-1. Example of running the script from the power shell.

A.2.1 Additional Notes

When running the script, take note of the following:

- If a run is unsuccessful, there will be error messages in the command prompt.
- The script creates a temp folder created in the input trajectory folder (dir_trj) named "mls_trj_idx_tmp". In order to re-run the script, delete that “tmp” folder. The folder in which the temporary geodatabase is stored can be helpful for troubleshooting when an error occurs. However, this “tmp” will be deleted if the script is fully executed.
- The process outlined above is not the only way to run the script. It can be run through VS-code or other IDE supporting python. However, when using these tools, ensure that the interpreter is set to be the python.exe installed with ArcGIS Pro.

A.3 QUERIES TO IDENTIFY DATASETS

The output mobile lidar trajectory indexing file (in csv format) contains the following fields:

- KML: Input KML file name.
- MLS: Mobile lidar file segment/name from the KML file.
- HWYNAME0: Highway name from milepost geodatabase matching the starting point of the mobile lidar trajectory. (Empty if there is no match)
- HWYNUMB0: Highway number from milepost geodatabase matching the starting point of the mobile lidar trajectory. (Empty if there is no match)

- MP0: Milepost number matching the starting point of the mobile lidar trajectory. (Empty if there is no match)
- LRM_KEY0: Linear Referencing Method Key from the milepost geodatabase matching the starting point of the mobile lidar trajectory. (Empty if there is no match)
- LRS_KEY0: Linear Referencing System Key from the milepost geodatabase matching the starting point of the mobile lidar trajectory. (Empty if there is no match)
- Dist2MP0: Distance between the starting point of the mobile lidar trajectory and its matching milepost point. (-1 if there is no match)
- HWYNAME1: Highway name from milepost geodatabase matching the end point of the mobile lidar trajectory. (Empty if there is no match)
- HWYNUMB1: Highway number from milepost geodatabase matching the end point of the mobile lidar trajectory. (Empty if there is no match)
- MP1: Milepost number matching the end point of the mobile lidar trajectory. (Empty if there is no match)
- LRM_KEY1: Linear Referencing Method Key from the milepost geodatabase matching the end point of the mobile lidar trajectory. (Empty if there is no match)
- LRS_KEY1: Linear Referencing System Key from the milepost geodatabase matching the end point of the mobile lidar trajectory. (Empty if there is no match)
- Dist2MP1: Distance between the end point of the mobile lidar trajectory and its matching milepost point. (-1 if there is no match)

To query the mobile lidar segments from the index table, this csv file can be opened in Microsoft Excel following these steps:

- Go to “Data” tab and activate “Filter” button.
- In the drop down menu under the HWYNAME, HWYNUM, LRM_KEY, or LRS_KEY columns (for both starting and end points), filter them based on the target highway index.
- Under MP0 and MP1, use the number filter option and pick between to define the milepost range for the query.
- Gather the MLS segments indicated in the KML and MLS fields.

	A		B		C	D	E	F	G	H	I	J	K	L	M	N
1	KML	MLS		HWYNA	H	MP0	I				HWYNA	-	MP1	LRM_KEY	LRS_KEY	Dist2M
4	G16011-3_segments	hwy001_NB_1_hwy001_NB_1_20170710_172042_1Profiler_zfs_2		PACIFIC	1	252.5	00100	74	PACIFIC	1	253.1	00100D00	000100100	52.85287		
5	G16011-3_segments	hwy001_NB_1_hwy001_NB_1_20170710_172042_1Profiler_zfs_3		PACIFIC	1	253.1	00100	57	PACIFIC	1	253.7	00100D00	000100100	44.01516		
9	G16011-3_segments	hwy001_NB_2_hwy001_NB_2_20170710_174412_1Profiler_zfs_2		PACIFIC	1	252.4	00100	29	PACIFIC	1	253	00100I00	000100200	55.92741		
10	G16011-3_segments	hwy001_NB_2_hwy001_NB_2_20170710_174412_1Profiler_zfs_3		PACIFIC	1	253	00100	49	PACIFIC	1	253.6	00100I00	000100200	42.37479		
11	G16011-3_segments	hwy001_NB_2_hwy001_NB_2_20170710_174412_1Profiler_zfs_4		PACIFIC	1	253.6	00100	27	PACIFIC	1	254	00100D00	000100100	39.43671		
14	G16011-3_segments	hwy001zNB_3_hwy001zNB_3_20170710_180726_1Profiler_zfs_2		PACIFIC	1	252.5	00100	55	PACIFIC	1	253.1	00100I00	000100200	38.63167		
16	G16011-3_segments	hwy001zNB_3_hwy001zNB_3_20170710_180726_1Profiler_zfs_4		PACIFIC	1	253.7	00100	48	PACIFIC	1	254	00100I00	000100200	30.46013		
19	G16011-3_segments	hwy001zNB_4_hwy001zNB_4_20170710_182920_1Profiler_zfs_2		PACIFIC	1	252.4	00100	36	PACIFIC	1	253	00100I00	000100200	49.28864		
23	G16011-3_segments	hwy001zSB_2_hwy001zSB_2_20170710_173447_1Profiler_zfs_1		PACIFIC	1	253	00100	36	PACIFIC	1	253.5	00100D00	000100100	41.27704		
24	G16011-3_segments	hwy001zSB_2_hwy001zSB_2_20170710_173447_1Profiler_zfs_2		PACIFIC	1	252.4	00100	31	PACIFIC	1	253	00100I00	000100200	18.20215		
28	G16011-3_segments	hwy001zSB_3_hwy001zSB_3_20170710_175816_1Profiler_zfs_1		PACIFIC	1	253.4	00100	58	PACIFIC	1	252.9	00100I00	000100200	43.35511		
29	G16011-3_segments	hwy001zSB_3_hwy001zSB_3_20170710_175816_1Profiler_zfs_2		PACIFIC	1	252.2	00100	62	PACIFIC	1	252.8	00100D00	000100100	77.8045		
33	G16011-3_segments	hwy001zSB_4_hwy001zSB_4_20170710_182022_1Profiler_zfs_1		PACIFIC	1	252.8	00100	33	PACIFIC	1	253.4	00100D00	000100100	36.96488		
34	G16011-3_segments	hwy001zSB_4_hwy001zSB_4_20170710_182022_1Profiler_zfs_2		PACIFIC	1	252.8	00100	17	PACIFIC	1	252.2	00100I00	000100200	23.67943		
38	G16011-3_segments	hwy001zSB_5_hwy001zSB_5_20170710_184008_1Profiler_zfs_1		PACIFIC	1	252.9	00100	46	PACIFIC	1	253.5	00100I00	000100200	37.17686		
39	G16011-3_segments	hwy001zSB_5_hwy001zSB_5_20170710_184008_1Profiler_zfs_2		PACIFIC	1	252.4	00100	77	PACIFIC	1	252.9	00100D00	000100100	22.6574		

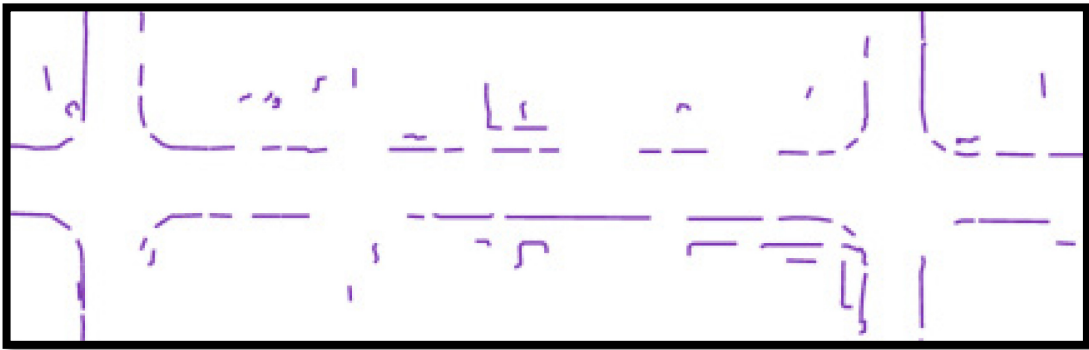
Figure A-0-2. Example of data query using the milepost indexing table (HWYNAME: PACIFIC, MP: 252 - 254).

APPENDIX: B

APPENDIX B: USER MANUAL CURB EXTRACTION

SPR850 Data Preprocessing Tools for Curb Extraction

V 1.0 3/1/2024



*Developed by Ezra Che and Michael Olsen
Oregon State University*

B.1 OVERVIEW

This custom package of tools is designed to extract curbs from mobile lidar datasets. The primary input is a folder with las/laz files to process. The primary outputs include a las/laz file with ground points, curb points, and road surface extracted, a simplified las/laz file with curb points, and a bsq raster of the curbs for input into GIS. The output classified las/laz can then be used as input to the ROME tool to improve the bike lane and median extraction. Although recommended for improved results, it is not necessary to complete this process to use the ROME tool. These tools require a computer with significant processing capabilities (e.g., CPU, disk space, RAM) to perform efficiently.

There are two scripts to support this process. The first script executes several software tools (Table B-1) in sequence for the extraction process including ground filtering, curb detection, and rasterization. It is a batch script that can be run from windows explorer after the data paths are set correctly. The script will automatically determine the filenames and locations of the outputs of the previous step for input into the next step when calling the executables. The second script is an ArcPy script run in ArcGIS software. This script converts the output bsq raster files into polylines within a geodatabase using several GIS operators (Table B-2).

Table B-1. Executables called for the curb extraction process*.

Executable	Description	Outputs
EZPC_Create_Lst_File.exe	Creates a text file listing all las/laz files within a directory.	.txt file
EZVox_Ground_Filtering.exe	Performs ground filtering of the las/laz dataset to classify points as ground (2) or non-ground (1). For more details on the ground filtering process please see Che et al. (2021).	Las/laz file with ground points classified (2).
EZFeat_Curb_Detection.exe	Performs road surface and curb extraction of the point cloud data. It inputs the ground points as a starting point. Curbs are identified as areas with abrupt changes in elevation adjacent to the road surface within specified tolerances. For more details please see Che and Olsen 2019.	Las/laz file with curbs only Las/laz file with all points but ground, curbs and road points classified
EZVox_Bin_N_Grid.exe	Bins points into grid cells and computes statistics within those grid cells.	bsq (raster) file

**Note that these tools are provided by EZDataMD LLC.*

Table B-2. ArcPy Script functions

Function	Description	Link
Raster Calculator	The raster calculator converts all cells with data to a value of 1 and all other cells as 0. It produces an integer raster. Calculation: Int(Con(IsNull(Input_raster),0,1))	https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/raster-calculator.htm https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/int.htm https://pro.arcgis.com/en/pro-app/latest/tool-reference/image-analyst/conditional-evaluation-with-con.htm
Thin	This function then thins the pixels in the raster to be 1 cell wide.	https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/thin.htm
Raster to Polyline	This function then connects the pixels to draw polylines.	https://pro.arcgis.com/en/pro-app/latest/tool-reference/conversion/raster-to-polyline.htm
SimplifyByStraightLinesAndCircularArcs	This function then simplifies polyline to minimize jagged effects as well as creates circular arcs to replace several line segments in areas with substantial curvature.	https://pro.arcgis.com/en/pro-app/latest/tool-reference/editing/simplifybystraightlinesandcirculararcs.htm

Script #1 Instructions

To simplify the extraction process, a batch script is provided. There are two versions: One for international feet and one for meters where the parameters are adjusted. Through ODOT SPR866, these parameters were fine-tuned and shown to be robust for many different datasets.

The following steps are needed to run the batch script:

1. If not preformed previously, run the Microsoft Redistributable which supplies the various system dlls needed: <https://learn.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-170> This is the direct link to the installer:
https://aka.ms/vs/17/release/vc_redist.x64.exe
2. Extract the SRP850datapreprocessing.zip file to a desired location (e.g., “C:\lidar”)
3. Open the .bat file in a text editor to edit.
4. Change the program folder (*program_folder*) to the location where the executables are stored. (Note that in batch scripts there should not be spaces around the “=”).
5. Change the input folder (*input_folder*) to the location where the las/laz files to be processed are located.
6. Change the output folder (*output_folder*) to a location where you would like the output files to be stored.
7. Save the script and run it by double clicking it. It may take several minutes to run.
8. Alternatively you can run the script by opening the PowerShell and dragging and dropping the .bat file into the shell.
9. Verify that no errors were triggered and the outputs are correct by browsing the files and importing into ArcGIS Pro.
10. For data to be georeferenced properly in ArcGIS Pro with correct units, you must set it up to read information from the world file if this has not been done before. To do this: Go to the *Project* tab, click *Options*, select *Raster and Imagery*, expand the *Raster Dataset* Category, and Check the *Use world file to define the coordinates of the raster* button. Hit *ok* to close the window and save the setting.

Script #2 Instructions

The second script is run within ArcGIS Pro. Note that depending on the version of ArcGIS pro, some modifications to the script may be needed.

The following steps are needed:

1. Import the toolbox with the script into ArcGIS Pro.
2. Import the bsq files into ArcGIS pro from the [output_folder\curb_dem](#). Note that there are several bands available with different information stored in each.
3. Use the “*Define Projection*” tool to set the spatial reference of the BSQ file to match your input data.
4. Run the python script through the Geoprocessing subwindow. There are 6 parameters that need to be specified:
 - a. *Input raster* (Select the raster layer you wish to convert)
 - b. *BandName* (Defaults to Z but can be changed to another band, if desired).
 - c. *Workspace Location* – the geodatabase to write the output layers to.
 - d. *Output coordinate system* (match the spatial reference in step 3). This is optional. This will reproject the data unless a projection has not been specified for the data. If that is the case, it will assign the coordinate system to the data without reprojecting.
 - e. *Smoothing Tolerance* (Set to a desired tolerance for the accuracy curb lines. A higher value will result in smoother, more linear lines with less jagged features, but may also miss some complexities).
 - f. *Cell size*. This normally should be set to be equivalent to the cell size used in Script #1. The default value is 0.1 m or 0.328 ft.
5. Review the output polylines (will be added to the map automatically). If necessary, repeat the script with a different smoothing tolerance. Note that there will be some false positives that will require manual editing.

The screenshot shows the 'Geoprocessing' window in ArcGIS Pro, specifically the 'CurbLines' script. The 'Parameters' tab is active, and the 'Environments' tab is also visible. The parameters are as follows:

Parameter	Value
Input Raster	T06_HWY091_MP0-20_SB_20200605_173727_Profiler.zfs_28_grnd_curb_ezft_avg.bsq
Band Name	Z
Workspace Location	ODOTSPR850data.gdb
Output Spatial Reference	OCRS_Portland_NAD_1983_2011_LCC_Feet_Intl
Smoothing Tolerance	0.5 International Feet
Cell_Size	0.328 International Feet

Figure B-0-1. Running the second script to generate curb polylines from the rasters.

The script can be run in **batch mode** for several rasters by right clicking the script in the catalog and selecting batch. *Note that all files need to be in the same coordinate system or the coordinate system parameter should be left blank.*

Geoprocessing

Batch CurbLines2

Parameters Environments

Batch Input Raster

- 1_T01_HWY068_SB_20200603_173632 Profiler.zfs_18_manual_label_grnd_curb_ezft_avg.bsq
- 2_T01_HWY068_SB_20200603_173632 Profiler.zfs_26_manual_label_label_grnd_curb_ezft_avg.bsq

Band Name

Z

Workspace Location

testdataset.gdb

Output Spatial Reference

OCRS_Portland_NAD_1983_2011_LCC_Meters

Smoothing Tolerance

1 Meters

Figure B-0-2. Running the second script in batch to generate curb polylines from the rasters.

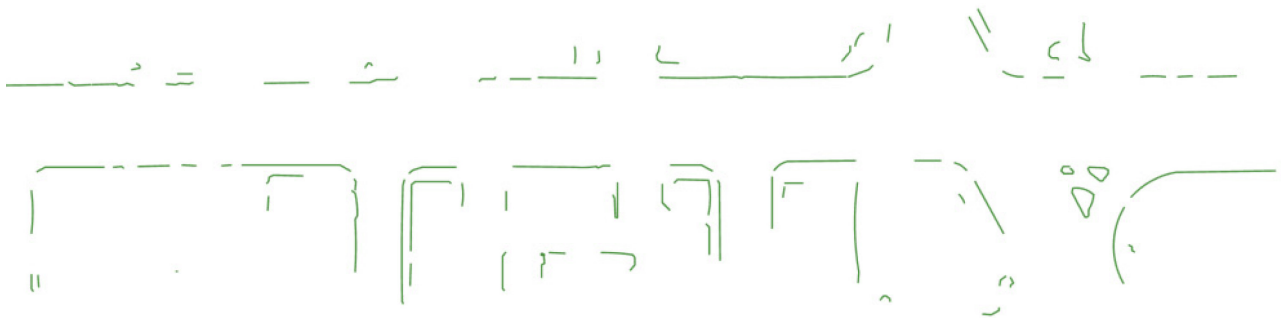


Figure B-0-3. Example output feature class of curblines in ArcGIS Pro.

APPENDIX: C

APPENDIX C: USER MANUAL: ROME

User Manual

Road Object and Marking Extractor v3.7



Developed By:
Jaehoon Jung, PhD
Assistant Professor
Department of Urban Engineering
Gyeongsang National University
jhjung@gmail.com

C.1 DISCLAIMER

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR

OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

C.2 INTRODUCTION

Welcome to the user manual for the ROME (Road Object and Marking Extractor) Program!

Many state Departments of Transportation (DOTs) track several metrics for compliance of pavement markings, such as appearance and retro-reflectivity, which are analyzed and used in creating a plan of action for maintenance. Unfortunately, issues arise due to the timing and frequency of the data acquisition. Often, individual hand-held reflectometer readings are required to recheck compliance quantitatively, which is risky, due to requiring roadside operations, and labor-intensive. Mobile lidar scans of state highways are now being conducted on a regular basis by many DOTs, providing a single dataset that can support many applications more safely and efficiently than conventional methods. In addition to capturing geometric information along the highways, a side product of lidar scans is intensity (return signal strength), which has a great potential for estimation of retro-reflectivity. The mobile lidar data is also useful to help identify the locations and extents of pavement markings for asset management. These road markings, in turn, help identify important multi-modal transportation assets such as bike lanes, medians, and crosswalks.

ROME is a user-friendly, automated tool designed for extracting and assessing various road features from mobile lidar data. The inputs to the program are point cloud (ASPRS LAS v1.2 format) and trajectory (asciitri) data obtained by a mobile lidar unit. The outputs include GIS shapefiles with road marking extraction and retroreflectivity evaluations that can be utilized for informed decision-making for the maintenance of various road objects. The ROME tool contains a simple interface that does not require the user to have extensive knowledge of the program in order to successfully run it. ***At this time, road objects should be able to be extracted from most mobile lidar systems; however, the radiometric calibration applied for retro-reflectivity evaluation is specific to the Leica Pegasus: Two mobile lidar system and will not produce correct results for other systems.*** Future releases may address these when radiometric calibrations have been completed and are integrated into the software. Please contact Jhjung1216@gmail.com for more information on this process.

Please note that some of the material in this section has been adapted from a final report submitted to the research sponsor, Oregon DOT SPR-799, of the previous version of the RoME tool (v1.0).

C.3 INTENDED AUDIENCE

This tool is designed for maintenance personnel, engineers, and other authorities in transportation departments who need this information to help make appropriate decisions for the evaluation of road markings and pedestrian facilities. It is assumed that users have basic knowledge of road markings and pedestrian facilities as well as an understanding of appropriate uses of mobile lidar data, as a function of spatial accuracy, resolution, and other variables. To learn more about mobile lidar and its capabilities, please visit the e-learning website:

www.learnmobilelidar.com.

C.4 PARTICIPANTS

Dr. Jaehoon Jung has been the principal developer across the series and currently maintains the software. RoME v1.0 was initially developed at the Civil Engineering Geomatics Lab at Oregon State University, with support from the Oregon Department of Transportation (ODOT) for SPR799. At the time, the tool was named the Road Marking Extraction or RoME. The Oregon Department of Transportation (ODOT) provided several of the datasets and valuable input throughout the process. A listing of these personnel is available in the ODOT SPR-799 final report. OSU provided support for the development of the GUI through the NSF I-Corps program. The development of RoME v1.0 and its associated algorithms was a collaborative effort involving Jaehoon Jung, Erzhuo Che, Michael J. Olsen, Christopher Parrish, and Joseph Greenwood. These contributors were all members of the Geomatics research group within the School of Civil and Construction Engineering at Oregon State University.

Pactrans provided funding for the development of RoME v2.0. Subsequent versions, RoME v2.1 to 2.13, were developed by Jaehoon Jung, which expanded the capabilities of RoME. Drs. Olsen and Che continued to provide assistance. These versions substantially improved the accuracy of the extraction as well as the efficiency.

Version 3.0 substantially expanded the capabilities and was renamed the Road Object and Marking Extraction tool, or ROME given its ability to extract more features. Dr. Jaehoon Jung lead the development of the algorithms and wrote the code. Drs. Che, Olsen, Parrish and Turkan OSU have assisted with algorithm development and workflow. Drs. Che, Olsen, and Rastiveis (OSU) performed rigorous testing of the software. Additional features were added to ROME by Dr. Jaehoon Jung, which was partially funded from Oregon DOT through SPR850 and SPR866. These improved clustering of road marking and lane extraction. This work resulted in detailed annotation of point clouds and the training of a deep learning model using those point clouds. These features and deep learning model were integrated into ROME 3.0 and above. Starting September 1, 2023, Dr. Jaehoon Jung continued developing ROME in his new position at Gyeongsang National University, providing numerous feature improvements and updates to the software, including substantially improved extraction results, data export capabilities of these new features with attribution, and a much-expanded GUI. Version 3.7 is the latest.

ROME continues to be developed and maintained by Dr. Jaehoon Jung, now based at Gyeongsang National University.

C.5 PUBLICATIONS

Several research publications and reports outline the algorithms and procedures used by ROME, including the following:

- Jung, J., Olsen, M.J., Che, E., Parrish C., and Turkan, Y. (In Press) “Instance-based clustering of road markings with wear and occlusion from mobile lidar data.” *Journal of Computing in Civil Engineering*

- Jung J., Olsen, M.J., Che, E., and Parrish C. (2020). “Efficient extraction and evaluation of complex pavement markings from mobile laser scan data”, Pactrans Final Project Report.
- Che, E., Olsen, M.J., Parrish, C.E., and Jung. J. (2019). “Pavement Marking Retroreflectivity Estimation and Evaluation using Mobile Lidar Data.” *Photogrammetric Engineering and Remote Sensing*, 85(8), 573-583. ASPRS.
<https://doi.org/10.14358/PERS.85.8.573>
- Jung, J., Che, E. Olsen, M.J., and Parrish, C. (2019). “Efficient and Robust Lane Marking Extraction from Mobile Lidar Point Clouds”. *ISPRS Journal of Photogrammetry and Remote Sensing*, 147, 1-18.
<https://doi.org/10.1016/j.isprsjprs.2018.11.012>
- Olsen, M.J., Parrish, C.E., Che, E., Jung, J., and Greenwood, J. (2018). “Lidar for maintenance of pavement reflective markings and retroreflective signs”. Vol I. Reflective Markings, Oregon DOT SPR-799 Final Report.

C.6 GETTING STARTED

The ROME tool does not require installation. Double click “ROME.exe” to start extracting and evaluating road markings from mobile lidar data. It may take approximately 40 – 50 seconds until the GUI appears on your screen for an initialization process to complete. (Note that the initial run of the tool on a computer may require a few minutes to setup the program). Installing NVIDIA CUDA toolkit 11.2 may help speed up the initial setup and initialization of the GUI. The file “rome_2023-6-7-10-22-11_CNNVGG6_patch(41)_cell(0.1)_1-10-11-12-14-15-16-17-21-40-50_CNN.hdf5” is required in order to perform some point cloud classification tasks necessary for the extractions.

Please note that a set of ArcPy scripts are also available to convert the output of the ROME tool into a GIS geodatabase with additional analysis layers. For more details on these tools, please see the “GIS Database” section. These were delivered with SPR799.

C.7 LICENSING

ROME v1.2 is available to use at www.learnmobilelidar.com as a demonstration of the software, but does not perform the extraction as effectively and efficiently as the current version. It also does not extract the same amount of features. ROME v3.7 requires a license to run the software beyond evaluating the software for demonstration purposes only. Contact jhjung1216@gmail.com to discuss licensing options. ODOT is provided with a license for running ROME 3.7 as part of the deliverables of SPR850 in accordance with the research contract.

C.8 OVERVIEW

As an overview, the ROME tool first subdivides the point cloud data into sections at fixed intervals (run length). It then performs ground filtering. After ground filtering, the road surface is extracted from the data. Subsequently, road markings in the point cloud datasets are extracted

and classified using a custom rule-based approach in conjunction with a deep learning convolutional neural network (CNN) model. Utilizing the road markings as a boundary, it then extracts bike lanes and key pedestrian facilities such as medians and crosswalks.

For the markings, there are three primary steps: 1. Extraction- identification of higher intensity locations corresponding to markings in the road surface, 2. Clustering- combining smaller segments of a marking to form a single cluster, and 3. Classification- utilizing the rule-based approach with CNN in order to determine what type of marking is represented. These building blocks then form the formation of the GIS polyline outputs of the tool.

For enhanced bike lane extraction, the user should run the EZFeat Curb Extraction tool prior to running ROME 3.7. A batch file with a streamlined workflow for executing this tool is provided with the SPR850 deliverables.

C.9 TOOL FEATURES

Several features contained within the Graphical User Interface (GUI) of the ROME program are shown in Figure C.1 and C.2 and described in Table C.1.

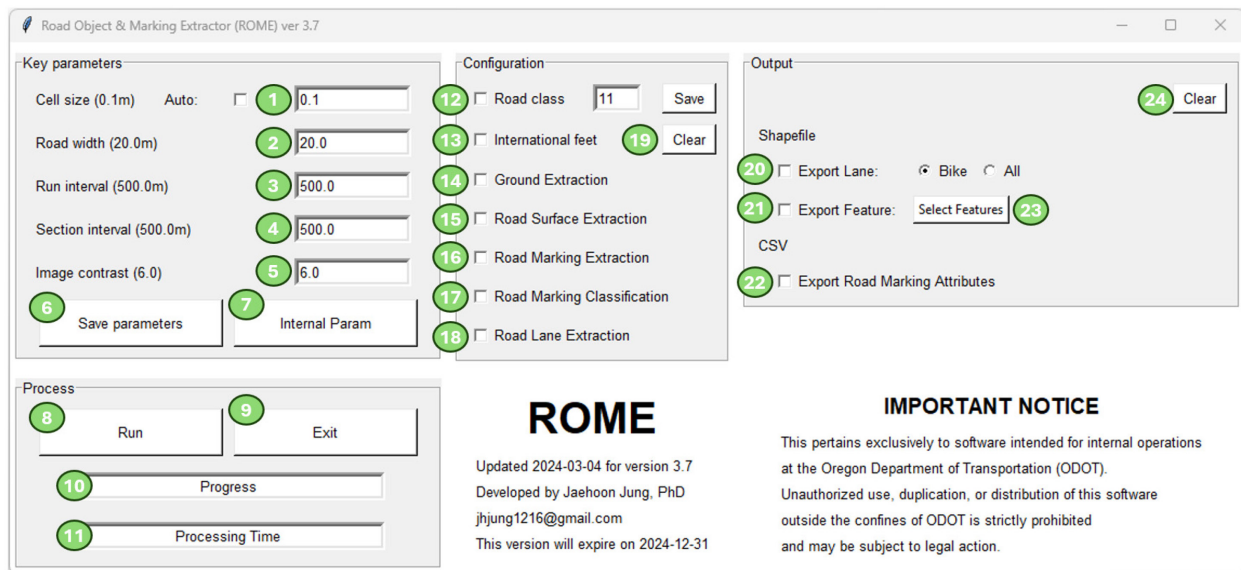


Figure C-0-1. Main GUI of the ROME program (version 3.7)

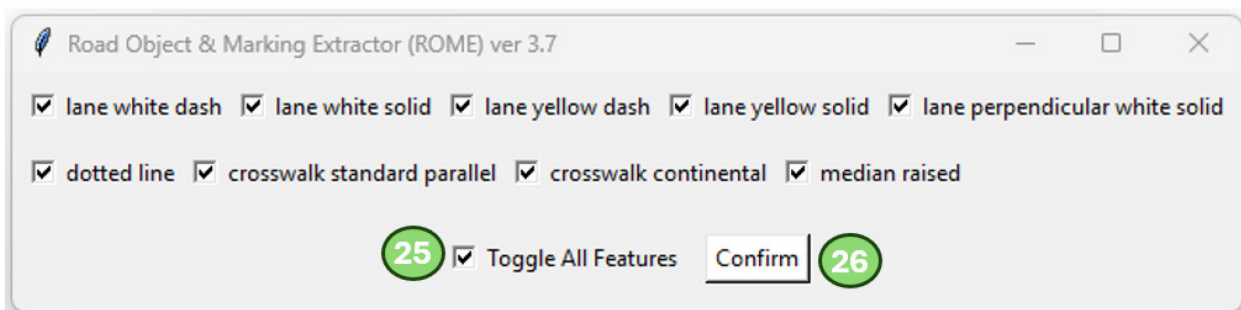


Figure C-0-2. Interface for selecting features for GIS layer outputs

Table C-1. Description of the features in the main GUI shown in Figure C-1 and C-2

No.	Description
1	Here the user selects the grid cell size used for a rasterization process used in the extraction. The grid cell size is recommended to be smaller than the width of lane markings to be detected but should not be too small to reduce processing time or result in substantial unoccupied grids on the rasterized image. It is also dependent on the resolution of the mobile lidar data, which is a function of vehicle speed. In general, 0.1m has worked well on testing of numerous mobile lidar datasets. Select the 'Auto' checkbox to enable the tool automatically to determine the cell size based on the point density estimated by the trajectory data.
2	This is the estimated road width to be processed for road marking extraction. If the value is too large, more false positives will occur. In general, 20.0 meters covers most typical highways worldwide and has been set as the default value.
3	This is the run interval value to divide a large road into smaller subsets (run) to improve processing efficiency. 500m has been set as the default value.
4	This is the section interval value to divide the road data into smaller sections for detailed line fitting for the retro evaluation and GIS layer generation. 500m has been set as the default value.
5	The image contrast parameter can be adapted to reduce the false negatives. The larger the values, the more the false negatives are reduced. However, too large of a value may also increase the number of false positives. In general, the contrast parameter is recommended to be 6.0, but optimal value may vary depending on different datasets.
6	This tool enables the user to save and apply the current key parameter set for the run. The changes in the key parameter set will also be applied to the internal parameter set.
7	This tool enables the user to change the internal input parameters. A new window will appear and allow the user to change the parameters (See the Appendix for more details). Note that it is generally recommended only for the user who has sufficient experience and knowledge of the program to change the internal input parameters.
8	This tool enables the user to browse to the folder containing the mobile lidar data and select them to run the road marking extraction and feature extraction. For classification, the user also needs to select a deep learning model (*.hdf5). This file is located in the same directory as the ROME executable. <i>A corresponding trajectory data (asciitry) must be copied to the same folder where the input LAS files are located. The input LAS files must contain the name of the trajectory data in their file names. The user can select any file from the folder, then the tool will automatically sort and process all the LAS or LAZ files in the folder.</i> At the end of the process, all the output SHP files will be placed in a new subfolder created by the tool under the directory where the input files are located. The date and time that a new process started will be assigned to the new folder name so that multiple runs will be identifiable.
9	This button enables the user to exit the program.
10	Displays the progress.
11	Displays the processing time.

No.	Description
12	Select the checkbox if the input las files already have the road surface classified. If selected, ROME will use the classification field of LAS file to identify road surface determined previously in another software package. The user can specify the specific road surface class (11 is default). Be sure to click ‘save’ button when the value is updated. If the checkbox is not selected or if road surface class does not match the actual road surface points, the tool will segment road surface using a built-in algorithm.
13	Select the checkbox to process the data in international feet rather than meters. Selecting the checkbox will also swap the parameter interface to appear in international feet. Note that the user should verify that they are consistent with using the same units throughout. If the input LAS file is in international feet, this box should be checked such that the output will be international feet. However, if the user provides a LAS file in meters but then selects international feet, erroneous results will occur.
14	Select the checkbox to extract ground surface, which is recorded with unique number (0 = non-ground surface, 2 = ground surface) in a new field “ground” of the resulting LAS or LAZ file. The user can specify the incidence angle in the internal parameter window (ground angle). For example, if the value is set to 45 degrees, the tool will remove any non-ground objects whose angle is greater than 45 degrees above the typical ground surface.
15	Select the checkbox to extract the road surface. If the road surface was extracted previously in another software, the user can enter the classification field used to identify the road surface. This defaults to 11.
16	Select the checkbox to extract road markings.
17	Select the checkbox to perform a classification of some road marking, which are used to identify some of the extracted features. <i>Note that the classification model was trained using roadway datasets acquired in Oregon with the Leica Pegasus: Two system and may not produce correct results for other systems.</i> For training a new classification model based on sensor specific data for improved results, please contact jhung1216@gmail.com .
18	Select the checkbox to extract road lanes. They are generated longitudinally at the midpoint between the solid lane markings.
19	This tool enables the user to deselect all the checkboxes in both the <i>Configuration</i> and <i>Output</i> sections.
20	Select the checkbox to export the road lanes in GIS shapefile format. The user can choose either bike lanes or entire road lanes. This functionality was developed for SPR866.
21	Select the checkbox to export the road features in GIS shapefile format.
22	This tool enables the user to select a list of road features to export in GIS shapefile format. A new window will appear and present the selection options as listed in Figure 2. By default, all road features are selected.
23	Select the checkbox to generate the CSV files. See the “Output files” section for detailed information about these files. Note that this option is provided for legacy

No.	Description
	support. It is required in order to run all of the python scripts to produce the GIS data layers from SPR799.
24	This button is a duplicate of #19 to deselect options.
25	Select the checkbox to toggle the selection of all road features.
26	This button enables the user to save and apply the current road feature selection.

C.10 INPUT FILES

The ROME tool inputs mobile lidar data (ASPRS LAS or LAZ v1.2) and the corresponding trajectory data (*asciitrj*). Typically, MLS systems collect a substantial number of points along the roadway. Road surface segmentation is an important prerequisite for reducing the false positives due to unwanted objects (e.g., buildings, signs, trees, cars, sidewalk etc.) while improving processing efficiency. The current version can identify the road surface which is recorded with unique number in the classification field of the input point cloud file. The user can also select feature 12 in Table C.1 to use a pre-segmented road surface if this process was performed previously and identify the classification code in the LAS/LAZ file. If there is no recorded classification for the road surface, the tool will segment road surface in the process.

Table C-2 shows an example format for the trajectory data file, where **TIME**, **X**, **Y**, **Z** are the necessary fields to run the ROME tool, and the other fields can be ignored. *While other mobile lidar datasets should work provided the information is input in the same format, they have not yet been tested.* For custom development, please contact jhjung1216@gmail.com.

Table C-2. Example format for the trajectory data file (asciitrj)

TIME	X	Y	Z	PITCH	ROLL	HEADING	ST.DEV POS	ST.DEV. ANGLES	QUALITY
147179.190	57612.464	61356.273	73.863	1.860	0.844	234.282	0	0	1
147179.195	57612.463	61356.273	73.863	1.861	0.844	234.282	0	0	1
147179.200	57612.461	61356.273	73.863	1.863	0.845	234.283	0	0	1
147179.205	57612.460	61356.273	73.863	1.865	0.845	234.283	0	0	1
147179.210	57612.459	61356.273	73.863	1.866	0.845	234.284	0	0	1
147179.215	57612.458	61356.273	73.863	1.868	0.844	234.282	0	0	1

C.11 OUTPUT FILES

The final output consists of a GIS shapefiles and CSV (comma separated values) with the extracted features.

C.11.1 GIS Shapefiles

Two shapefiles are output from the tool: *lanes* and *features*. Figure C.3 shows an example of the *features* layer. The lanes tool contains lines associated with each lane as well as the bike lanes. Note that the layer will be named bike lanes if that is selected instead of all. The features tool contains the output pavement markings, crosswalks, and medians. These output files can be

imported into GIS software directly for further visualization and processing. The ArcPy Scripts described below are no longer necessary unless the user wishes to perform the stripe color reclassification or the *retrograder.py* script to output condition scores for sections of the highway.

The *lanes* shapefile generated for each extracted feature and contains the following fields:

- *FID*- a feature ID, unique to each lane segment.
- *LaneID* – a unique identifier for each extracted lane. (Note that multiple segments of the same lane can be linked together via this field).
- *Width* – estimate of the lane width.

The *features* shapefile (Table C.3) is generated for each extracted feature and contains:

- *StripeID* – a unique identifier for each extracted stripe.
- *SectionID* - the corresponding section to link the *Stripe* to the *Section* table.
- *NodeStart, NodeEnd* – The starting and ending indices of the nodes of the stripe.
- *Color* – The estimated color (white or yellow) of the stripe using the RGB colors mapped to the point cloud. Note that this field may not be reliable due to the limitations of RGB values acquired by a mobile lidar unit.
- *Material*- A field in which the user can input the type of material associated with a marking, if desired.
- *Length* – The length of the extracted stripe (in meters or feet).
- *ConditionScore* – a Grade (A - F) assigned based on the median retroreflectivity value of the simulated retroreflectivity readings for the stripe and the color of the stripe. (See Table C.10 and SPR-799 for details).
- *RetroNumPts* - Number of the retro samples on the stripe.
- *RetroMin, RetroMax, RetroMedian, RetroAve, RetroStdDev* – summary statistics of the simulated retroreflectivity readings for the stripe.
- *NumPtsPC* – The number of lidar points composing the stripe in the point cloud,
- *IntMin, IntMax, IntMedian, IntAve, intStdev* summary statistics of the intensity values for the lidar points belonging to that stripe.
- *Width* – width of the stripe based on the user input parameter.
- *Stripe Type* – A classification to indicate which feature was extracted:

- 'lane white dash',
- 'lane white solid',
- 'dotted line',
- 'crosswalk standard parallel',
- 'crosswalk continental',
- 'median raised',
- 'bike symbol'

Table C-3. Example of fields and data provided in the feature shapefile

StripeID	SectionID	Node Start	Node End	Color	Material	Length	Condition Score	Retro NumPts	Retro Min	Retro Max
1	1	1	2	White	N/A	10.064	D	20	84.522	121.604
2	1	3	4	White	N/A	3.241	A	7	339.864	373.280
Retro Median	Retro Ave	Retro StdDev	Num PtsPC	Int Min	Int Max	Int Median	Int Ave	Int StdDev	Width	StripeType
101.274	103.402	10.515	2775	0.045	0.502	0.259	0.249	0.073	0.1	Lane
373.280	361.432	15.072	642	0.074	1.000	0.849	0.803	0.178	0.1	Left arrow



Figure C-0-3. Example screenshot of the output layers in a geodatabase in ArcGIS Pro.

C.11.2 CSV Files (Legacy)

CSV file output is provided for legacy support. Select feature 23 in Table C.1 to generate the CSV files of the *Run*, *Section*, *Stripe*, *Node*, and *Retro* tables in CSV format. These files are output as they were the format for the output for RoME v1.0. ROME 3.7 outputs these files to provide legacy support during the transition. (Each of these tables will be described below). The output tables are linked through a set of IDs such that the attributes can be efficiently stored but still available throughout the various data layers. The generated CSV files can be imported into commercial GIS software or analyzed by themselves. ***Note that the units of distance measurements are either meters or feet and are based on the user's setting in the ROME software.***

To extract markings and to evaluate the retro values (when applicable), the current version of the tool divides the road data into smaller sections and fits a line to each marking detected within a section. By default, the section interval is set to 500.0 m; however, the user can enter a new section interval into feature 4 in Figure C.1. ***Note that line fitting and retro evaluation may be incorrect for complex and curved lane markings. The user can check the Root Mean Square (RMS) for each line fitting in the Stripe table (see Table C.6) to identify these features.***

The *Run* table (Table C.4) is generated as a summary of the data with a separate record for each LAS file. It contains the following attributes:

- *RunID* – A unique ID for each run.
- *HWYNumber* - The highway number, which can be modified by the user in the resulting CSV file.
- *Date*- the acquisition date, which is useful for tracking pavement marking degradation with time.
- *SectionIDStart*, *SectionIDEnd* – The starting and ending indices of the sections for each run (las file).
- *StripeIDStart*, *StripeIDEnd* – The starting and ending indices of the stripes extracted for each run.
- *NodeStart*, *NodeEnd* – The starting and ending indices of the nodes extracted for each run.
- Several user parameters, described earlier, are provided in the table as well for reference.
- *Software Version* – The current software version.
- *File Name* – Input LAS file name for each run.

Table C-4. Example of fields and data in the Run table

RunID	HWY Number	Date	Section IDStart	Section IDEnd	Stripe IDStart	Stripe IDEnd	Node Start
1	H1	20170725	1	2	1	6	1
2	H1	20170725	3	4	7	62	13
Node End	Section Interval	Grid CellSize	Angle DiffDeg	Stripe Width	Road Width	Software Version	File Name
12	10	0.05	15	0.1	7.2	2.4	T04...
124	10	0.05	15	0.1	7.2	2.4	T05...

The *Section* table (Table C.5) is generated for each section where stripes are extracted. It contains the following attributes:

- *trajMidX*, *trajMidY*, *trajMidZ* - location of trajectory's mid spot (meters or feet),
- *StripeIDStart*, *StripeIDEnd* – The starting and ending indices of the stripes extracted for each section.
- *RunID* - the corresponding ID to link the section table to the *Run* table.

Table C-5. Example of fields and data in the Section table

SectionID	trajMidX	trajMidY	trajMidZ	StripeIDStart	StripeIDEnd	RunID
1	71521.710	54613.878	136.311	1	4	1
2	71513.559	54619.596	136.244	5	7	1
3	71505.359	54625.336	136.178	8	10	1
4	71497.208	54631.049	136.110	11	13	1
5	71489.012	54636.801	136.050	14	16	1
6	71480.813	54642.549	135.986	17	19	1

The *Stripe* table (Table C.6) is generated for each extracted stripe and contains:

- *StripeID* – a unique identifier for each extracted stripe.
- *SectionID* - the corresponding section to link the *Stripe* table to the *Section* table.
- *NodeStart*, *NodeEnd* – The starting and ending indices of the nodes of the stripe.
- *Color* – The estimated color (white or yellow) of the stripe using the RGB colors mapped to the point cloud. Note that this field may not be reliable due to the limitations of RGB values acquired by a mobile lidar unit.
- *RMS* – Root Mean Square of the line fitting for individual markings. The line fitting is required to evaluate the retro values. The user can use the RMS values to identify and remove the lines with large RMS values if using for retroreflectivity analysis to ensure correctness.
- *Length* – The length of the extracted stripe (in meters or feet).

- *ConditionScore* – a Grade (A - F) assigned based on the median retroreflectivity value of the simulated retroreflectivity readings for the stripe and the color of the stripe. (See SPR-799 for details).
- *RetroNumPts* - Number of the retro samples on the stripe.
- *RetroMin*, *RetroMax*, *RetroMedian*, *RetroAve*, *RetroStdDev* – summary statistics of the simulated retroreflectivity readings for the stripe.
- *NumPtsPC* – The number of lidar points composing the stripe,
- *IntMin*, *IntMax*, *IntMedian*, *IntAve*, *intStdev* summary statistics of the intensity values for the lidar points belonging to that stripe.
- *Width* – width of the stripe based on the user input parameter.

Table C-6. Example of fields and data provided in the Stripe table

StripeID	SectionID	Node Start	Node End	Color	RMS	Length	Condition Score	Retro NumPts	Retro Min	Retro Max
1	1	1	2	White	0.012	10.064	D	20	84.522	121.604
2	1	3	4	White	1.027	3.241	A	7	339.864	373.280
Retro Median	Retro Ave	Retro StdDev	Num PtsPC	Int Min	Int Max	Int Median	Int Ave	Int StdDev	Width	Class
101.274	103.402	10.515	2775	0.045	0.502	0.259	0.249	0.073	0.1	Lane
373.280	361.432	15.072	642	0.074	1.000	0.849	0.803	0.178	0.1	Left arrow

The Node table (Table C.7) is generated for each node (start and end points for each stripe). It contains the following attributes:

- *NodeID* – unique ID for each node
- *X*, *Y*, *Z* – the X, Y, Z coordinates (meters or feet) for each node,
- *StripeID* – the corresponding StripeID to provide the link to the Stripe table.

Table C-7. Example of fields and data provided in the Node table

NodeID	X	Y	Z	StripeID
1	71526.766	54612.420	134.245	1
2	71518.552	54618.135	134.179	1
3	71521.947	54611.244	134.295	2
4	71519.298	54613.112	134.270	2
5	71522.562	54606.341	134.399	3
6	71514.357	54612.146	134.334	3

Simulated retro-reflectivity readings are generated at a fixed distance (user parameter: Default 2 feet) along each stripe. These readings are simulated by extracting the lidar points within a window size with the dimensions of the active window of the Delta LTX handheld retroreflectometer. The results are stored in the *Retro* table (Table C.8), which contains:

- *RetroID*- unique ID for each simulated retroreflectivity reading.
- *X, Y, Z* – the X, Y, Z coordinates (meters or feet) for each simulated retroreflectivity reading,
- *StripeID* – a link to the corresponding stripe where the simulated retroreflectivity reading is obtained.
- *NumPtsPC* – The number of lidar points within the simulated retroreflectivity reading that were used for the retro evaluation. This can also serve as a quality control metric.
- *Retro10* – The 10th percentile retro value from the lidar points within that window based on the SPR-799 radiometric calibration for the Leica Pegasus: Two.

Table C-8. Example of the fields and data provided in the Retro table

RetroID	X	Y	Z	StripeID	NumPtsPC	Retro10
1	71526.685	54612.478	134.244	1	44	99.933
2	71526.276	54612.667	134.241	1	45	86.530
3	71525.868	54612.956	134.239	1	39	93.282
4	71525.460	54613.245	134.236	1	40	90.638
5	71525.052	54613.534	134.232	1	39	84.522
6	71524.644	54613.822	134.229	1	38	100.653

C.12 GIS DATABASE

ArcGIS Python (ArcPy) scripts, *csv2gdb.py*, *stripeEditor.py*, and *retrograder.py* are included in the ROME package. The *csv2gdb.py* script inputs the *csv* files output from the *ROME* software and converts them to feature classes in a new GIS geodatabase. The *stripeeditor.py* script provides a simple way to correct stripe coloring based on digitized yellow stripes. It also can screen any transverse markings. The *retrograder.py* script aggregates the retroreflectivity results into grades for larger sections. The GIS tool was developed using ArcGIS python (Arcpy), which is the preferred scripting language for ArcGIS. Upon export of the *csv* files from the ROME tool, the user would then run the python scripts in sequence to create a file geodatabase containing the results of each run (Table C.9 and Figure C.4). While each run will likely be an individual section of highway, the final results can then be merged with previous runs for inclusion in a larger layer using the “merge” geoprocessing tool available within ArcGIS, as necessary.

The *csv2gdb.py* script (Figure C.5) first imports the *csv* files into the geodatabase as tables. The tool creates a new geodatabase or can write to an existing geodatabase. It then creates point feature classes from the *Node*, *Section*, and *Retro* tables based on the *X,Y,Z* values in those tables. The tool then uses the information from the *Node* table to reconstruct the lines for the stripe table based on the ID links to represent the individual markings extracted from the ROME tool. It then reconstructs a polyline from the points in the section table to create a polyline feature class for the *Run* table. Note that the initial tables that are no longer needed are deleted to avoid confusion. (The *csv* files themselves are not affected by the script. Nevertheless, it is recommended to archive them). The script takes approximately 30 seconds per mile of data, but

will vary depending on the hardware used and if files are located on a network vs. the local machine).

Table C-9. Description of data layers produced with the python script tool

Script#	Data Layer	Description
1	Run	Creates a polyline feature class with a single polyline for each run which represents an approximate version of the trajectory for each stripe
1	Stripe	Creates a polyline feature class with individual polylines for each extracted stripe (or segment of stripe). Contains quality control information on the stripe.
1	Node	Creates a point feature class with the start and end nodes for each stripe. The primary purpose of this table is simply for creation of the stripes.
1	Section	Creates a point feature class with points along the trajectory for each discretized section in the ROME tool. These sections are connected to create a polyline feature for the run feature class.
1	Retro	Creates a point feature class with simulated retro-reading samples every 3 ft (or other user specified increment) along each stripe. Contains various statistics related to the retro-reading from the lidar point cloud.
2	Stripe_Tfill	Creates a new version of the Stripe Feature Class with transverse stripes removed and coloring corrected.
3	MP_Retro (user defined name)	Creates a polyline feature class that contains an aggregated condition score of the stripes located within the section of a certain increment (e.g., tenth of a mile) based on the Oregon DOT milepost layers. (Can use any mile increment provided the fields remain constant). This layer is meant to be used for regional or statewide asset management purposes.

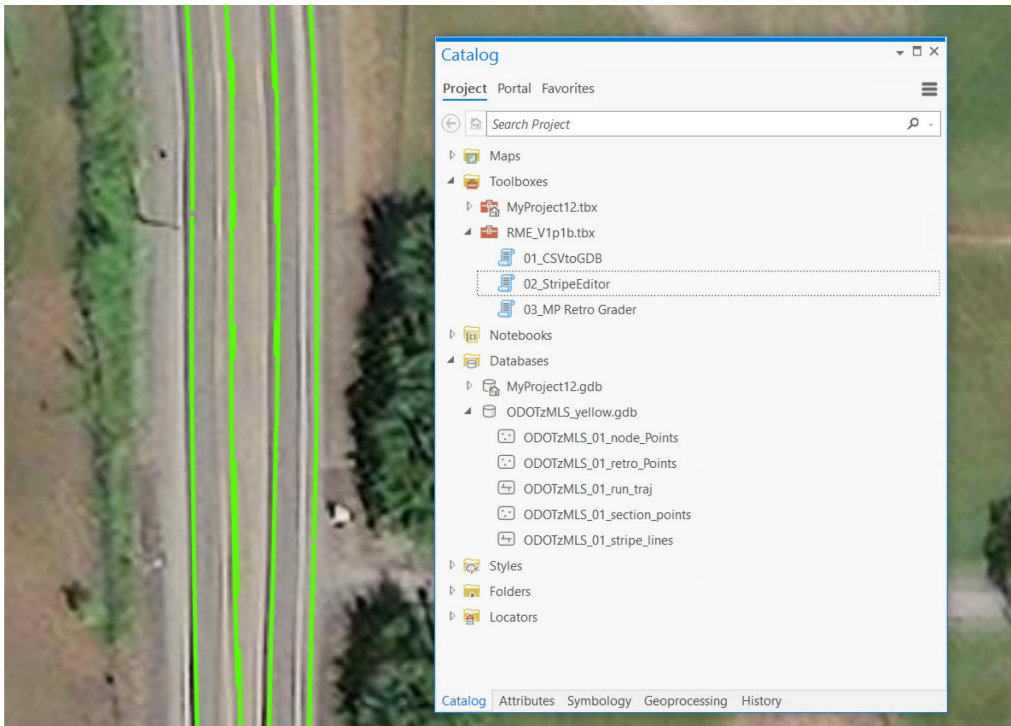


Figure C-0-4. Screenshot of the output layers in a geodatabase in ArcCatalog

In order to run this tool, the user must specify the following parameters:

- An input folder for the csv files. (e.g., C:\RetroData\hwy001\csv)
- An output folder for the geodatabase. (e.g., C:\RetroData\hwy001) – note that this folder should not be in the same folder as the csv files but can share the same root path.
- A name for the output geodatabase (e.g., hwy001retro). Do not include the “.gdb” as the script will automatically add that.
- The spatial reference of the data contained in the input csv files. (e.g., OCRS_SALEM_NAD_1983_2011_TM_Meters). Please make sure to input the coordinate system used for exporting the LAS files.

Example outputs are shown on the following pages. Figure C.5 shows the GUI for the csv2gdb.py tool in ArcGIS Pro and Figure C.6 shows an example of the extracted markings with attributes available in the *Stripe* table. ***Note that the input CSV filenames must not contain illegal characters except ‘_’.***

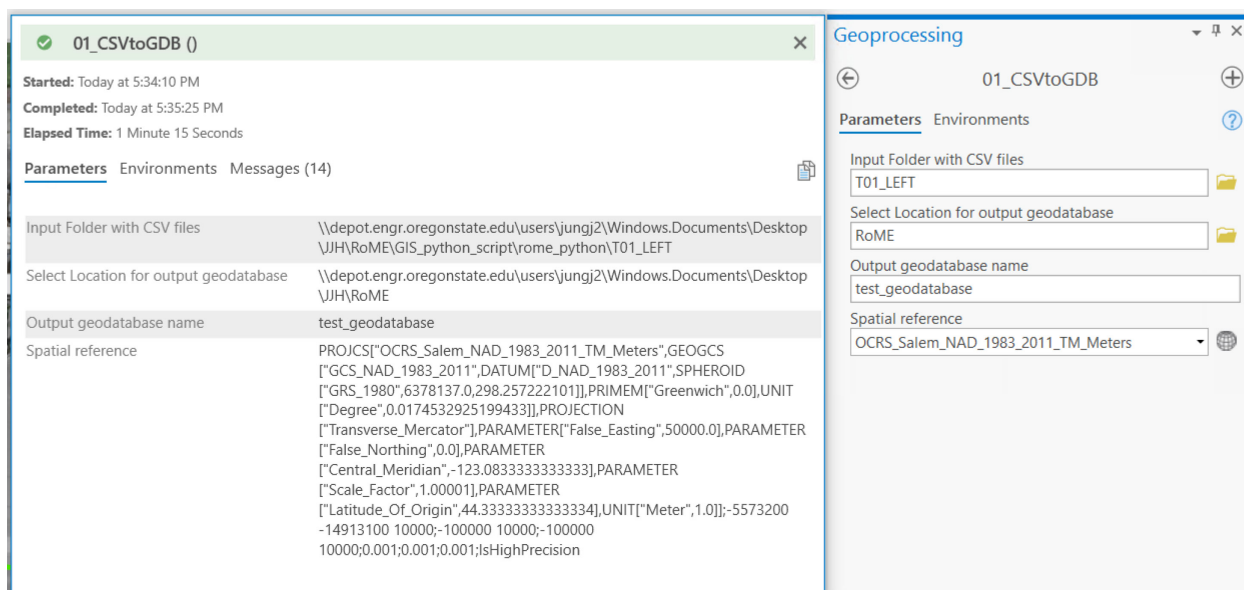


Figure C-0-5. GUI for the csv2gdb.py tool in ArcGIS Pro showing input parameters as well as command line output.



Figure C-0-6. Example output of the csv2gdb.py script showing attributes available in the stripe table.

A second (optional) python script stripeEditor.py inputs the stripe feature class as well as a feature class containing digitized lines of yellow stripes. The script changes the color of all lines within a user specified distance to be yellow and all others to be white. The assumption is that there are significantly fewer yellow lines, they are generally continuous, and thus can be quickly digitized. In cases where there are dashed yellow lines next to a solid yellow line, the user would still only need to digitize the main line since the dashed lines would be within the threshold

distance. In the case of a single dashed yellow line to indicate passing, the user could simply digitize a continuous line to represent the yellow line since this is only used as a reference for coloring, but the dashed lines would be preserved in the stripe Feature Class. However, a limitation of this approach resides in the fact that erroneous results would occur with a white line in close proximity to a yellow line.

The script also has an option to remove transverse markings (which are the bulk of the false positives of the ROME tool). If that option is selected, it creates a new stripe feature class with “_Tfill” appended to its name. Figure C.7 shows the GUI for the *stripeEditor.py* tool in ArcGIS Pro and Figure C.8 shows an example of the extracted markings with yellow and white stripes.

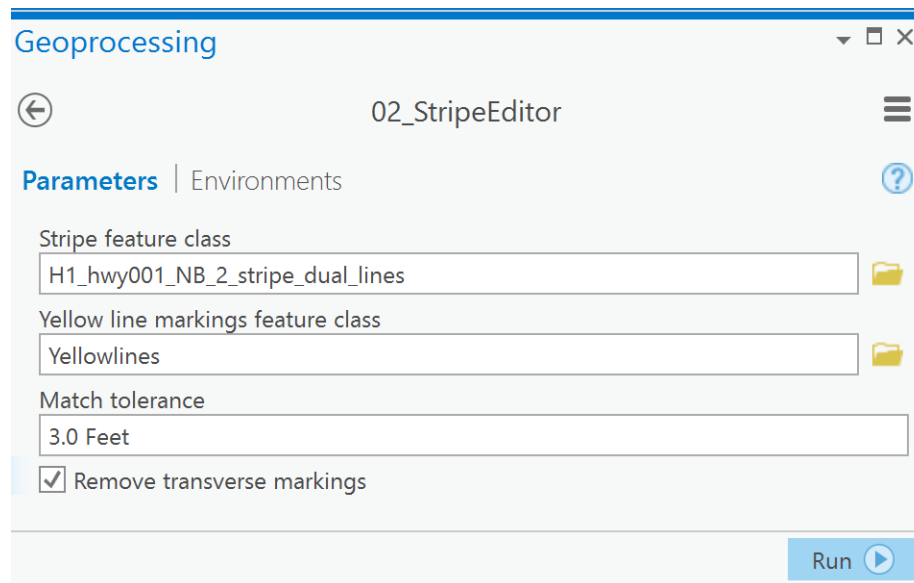


Figure C-0-7. GUI for the stripeEditor.py script.



Figure C-0-8. Example output of the stripeEditor.py script showing correct yellow and white stripes with transverse stripes removed.

To improve the data quality for asset management, it may be worthwhile to perform some manual cleanup of the data to remove erroneous stripes. The user may use some of the attributes such as stripe length, number of points, etc. as quality metrics to quickly select stripes that are likely false positives. They can then adjust the selection as appropriate and remove the false positives. They can also adjust the stripe color information by plotting the stripes with symbology in GIS and subsequently adjusting the color of the stripes to be correct. Note that the *retrograde.py* grades the stripes differently based on its color in accordance with specifications. As a shortcut, should a conservative result be desired for the aggregated asset management layer, one can change the color attribute field to be yellow for all stripes, which will result in a more conservative grading.

The *retrograder.py* script (Figure C.9) inputs a mile marker feature class (e.g., the 1/10th milepost layer) and then provides a retroreflectivity quality score for each 1/10th mile post section as an attribute field (labeled “MP”). The choice of the milepost layer depends on the resolution desired for asset management purposes.

The script reclassifies the condition score in the stripe table based on the median retroreflectivity value from all points in the retro feature class using the thresholds in Table 9. Although the ROME tool also performs this classification, the values are reclassified to account for any additional editing. Next, the script creates a buffer (e.g., 200ft) on each side from the polyline in the *run* feature class. It then clips the mile marker feature class to include only those points that lie within the buffer. (Note that all attributes are preserved such that the results can be linked back to the milepost layer if desired). The script then reconstructs a polyline between each mile marker and stores it in a new feature class. Each polyline is buffered and all *stripe* polylines from the *stripe* feature class located within that buffer are analyzed to provide the overall score

for that section. The grade for the section is assigned based on the thresholds in Table C-10 depending on the color of the stripe (white or yellow). The script computes the mean retroreflectivity grade of all stripes within each milepost section. (Note that for computation, the grades are converted to a numerical scale ranging from 0(F) to 5(A)). A value of Z represents no data.

Table C-10. Grade (Condition Score) thresholds

Grade	White	Yellow
A	>350	>250
B	250-350	200-250
C	150-250	125-200
D	100-150	100-125
E	50-100	50-100
F	<50	<50

The script contains the following input parameters (Figure C.9):

1. *Analysis Geodatabase* – The file path and name of the geodatabase output from the *csv2gdb.py* script.
2. *Stripe Feature Class* – The stripe feature class output from the *csv2gdb.py* script.
3. *Run Feature Class* – The run feature class output from the *csv2gdb.py* script
4. *Oregon DOT Tenth MilePost Feature Class* – The 1/10th milepost layer. (Another layer could be used if finer (e.g., 1/100th mile) or coarser (e.g., 1 mile) aggregate results are desired. It should be noted that this table must have a field named “MP”, which contains the milepost numbers of each marker. The program will reconstruct the polyline based on the order of the mileposts. The lower milepost is considered the start node of the section for the rating purposes). This ordering should be considered when field verifying the results since the direction of the mileposts may increase in the direction opposite the direction of travel.
5. *LRSKeyFilter* (optional) – A filter based on the LRS Key to filter the mileposts to a single route. This filter is useful for divided highways which have mileposts on different sides and areas where highways intersect to avoid misclassifying the results to the incorrect highway.
6. An *output name* for the output feature class providing retroreflectivity values at the milepost increments (e.g., hwy001_MP_Retro)
7. *Output layer spatial reference* (Optional) – Note that the spatial reference for the milepost layers is Web Mercator. As a result, there can be a significant amount of distortion with geoprocessing tools such as Buffer in ArcGIS, which can lead to undesired results of stripes not being considered in the analysis. The script allows the user to reproject the data layer to a low distortion coordinate system such as the

appropriate zone in the Oregon Coordinate Reference System (which will likely be used for las files and ROME output files). However, an alternative to reprojecting the milepost data is to simply use a larger buffer should the output be desired in Web Mercator. If this parameter is left blank, the program will not reproject the data.

8. *Buffer Distance* – Enables the user to specify a buffer distance. The default is 100 ft. This buffer distance should ensure that all stripes of interest are within that distance of the line formulated between the mileposts. The buffer distance is also used
9. *Delete temporary files* – By default the program will delete the various geoprocessing layers created to produce the final results. However, the user can uncheck this box for troubleshooting purposes.

An example output for a section of Interstate-5 in Salem area is shown in Figure C.10. This figure includes data from a single northbound and a single southbound pass from the center lane on each side.

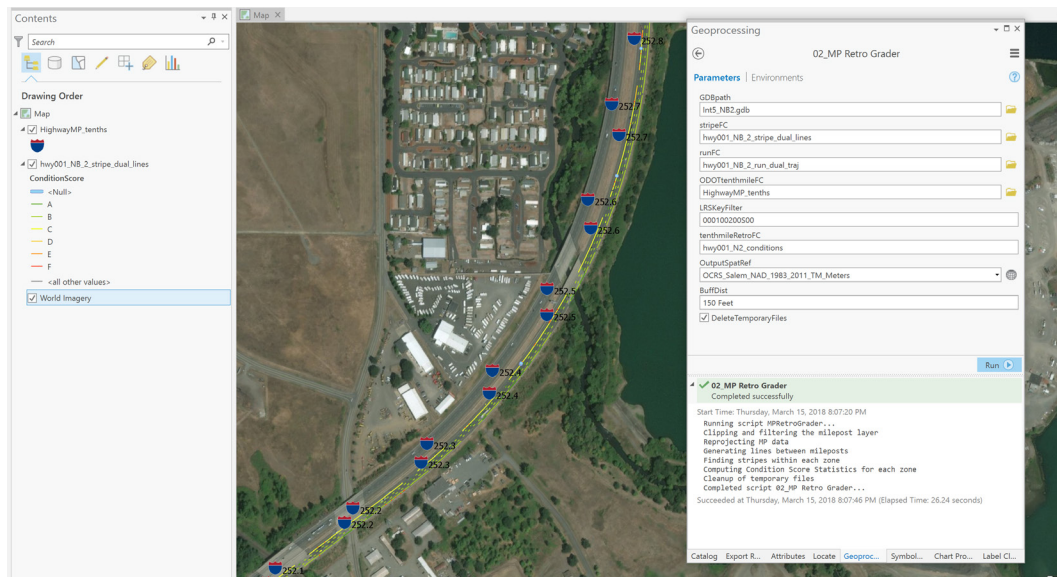


Figure C-0-9. GUI for the retrograder.py script and associated inputs in ArcGIS Pro

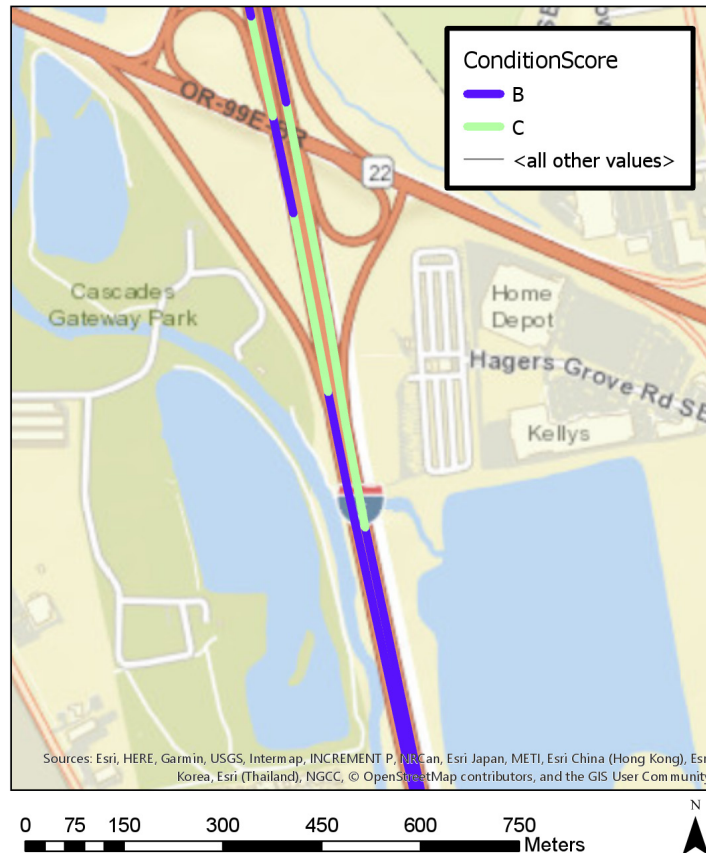


Figure C-0-10. Example output of the retrograder script showing the MP_retro results for a single northbound and single southbound pass colored by overall condition score.

C.13 RECOMMENDATIONS

There following tips are recommended to use the developed program effectively. They include:

- The classification and retro evaluation of the ROME tool (v3.7) were trained with data acquired by the Leica Pegasus Two MLS system. To improve the classification results on data collected with another sensor, the models need to be retrained with new datasets specific to the sensor. For custom classification, please contact: jhjung1216@gmail.com
- The retro evaluation for some complex road marking can be incorrect due to line fitting. It is recommended to refine the results using the RMS field provided in the Stripe table when necessary.
- The classification of yellow and white markings can sometimes be incorrect due to shadow, occlusion, lighting condition, and misalignment between the lidar and camera sensors. The Stripe Editor tool can help correct some of these issues, but manual updating may be necessary.

C.14 REFERENCES

ROME utilizes several open-source algorithms for data processing, including:

- cc3d: connected components on multilabel 3D images. <https://pypi.org/project/connected-components-3d/>
- Focal-loss: a loss function generalizes multiclass SoftMax cross-entropy. https://github.com/artemmavrin/focal-loss/blob/master/src/focal_loss_categorical_focal_loss.py
- Glob: Unix style pathname pattern expansion. <https://docs.python.org/3/library/glob.html>
- Laspy: Python library for lidar LAS/LAZ IO. <https://laspy.readthedocs.io/en/latest/>
- Math: mathematical functions. <https://docs.python.org/3/library/math.html>
- Matplotlib: Visualization tool with Python. <https://matplotlib.org/>
- Natsort: Simple yet flexible natural sorting in Python. <https://pypi.org/project/natsort/>
- NumPy: the fundamental package for scientific computing with Python. <https://numpy.org/>
- Open3d: open-source library that supports rapid development of software that deals with 3D data. <https://pypi.org/project/open3d/>
- Opencv-python: pre-built CPU-only OpenCV packages for Python. <https://pypi.org/project/opencv-python/>
- OS: Miscellaneous operating system interfaces. <https://docs.python.org/3/library/os.html>
- Patchify: A library that helps you split image into small patches. <https://pypi.org/project/patchify/>
- Pandas: open source data analysis and manipulation tool. <https://pandas.pydata.org/>
- Pyshp: Pure Python read/write support for ESRI Shapefile format. <https://pypi.org/project/pyshp/>
- Scikit-image: a collection of algorithms for image processing. <https://scikit-image.org/>
- Scipy: Fundamental algorithms for scientific computing in Python. <https://scipy.org/>
- TensorFlow: An end-to-end open source machine learning platform. <https://www.tensorflow.org/>

- TiffFile: A Python library to store NumPy arrays in TIFF. <https://pypi.org/project/tiffFile/>
- Time: Time access and conversions. <https://docs.python.org/3/library/time.html>
- Tkinter: python interface to Tcl/Tk. <https://docs.python.org/3/library/tkinter.html>

C.15 APPENDIX

This appendix (Figure 11) contains the full options available in the internal parameter window to operate the ROME tool (v3.7). Note that it is generally recommended only for the user who has sufficient experience and knowledge of the program to change the internal input parameters. Please see the above references for more information on these parameters. For any questions, please contact jhjung1216@gmail.com

Internal Parameters

Road surface extraction

Cell size for rasterization (0.1m)

Voxel size for sub-sampling (0.2m)

Ground angle (45deg)

Cell size for region-growing (0.2m)

Road width (20.0m)

Run interval (500.0m)

Road marking extraction

Image contrast (6.0)

Intensity threshold (0.1)

Save and exit

Retroreflectivity evaluation

Section interval (500.0m)

Stripe width (0.1m)

Stripe Length (0.5m)

Interval (0.5m)

Search window length (0.2m)

Search window minimum # points (5)

Search window width (0.045m)

Pecentile (10%)

Retro model coefficient 1 (1.19261)

Retro model coefficient 2 (373.28)

Figure C-0-11. Internal parameters for ROME v3.7

APPENDIX: D

APPENDIX D: TOOL EVALUATION RUBRIC

Table D-1. Summary of datasets explored for the evaluation.

Dataset Tested	# points	Length of highway	Types of objects present	Processing Time (seconds)	Comments

Modify the following table to reflect your evaluation results. Sample comments are provided for reference as to how to classify your comments. Delete any non-applicable comments. It would be ideal to complete a separate table for each tool tested. Be as specific as possible (e.g., the EZ Feat Curb Extraction Tool missed the curb in the areas of occlusion for the mobile lidar but captured most other areas).

Table D-2. Rubric for the evaluation.

Evaluation Category	Major Issue	Minor Issue	Positive Results	Comments
Tool stability	- Tool doesn't run. - Frequent unexplained crashes	Occasional or predictable crash	Tool runs as expected with minimal issues.	Note any situations where the program did not operate as expected.
Extraction Results	- No objects extracted. - Too many false positives that one can't make sense of the results.	- Some objects missed. - Some false positives observed.	- Overall most objects detected - Few false positives - Clear reasons why an object is missed (e.g., occlusion in the data) or	Comment on any specific issues you see with extraction results (e.g., in areas of occlusion, poor marking quality, etc.)

Evaluation Category	Major Issue	Minor Issue	Positive Results	Comments
			why a false positive is generated.	
Parameters	- Parameters are unclear and overwhelming. - Could not obtain good results with tweaking parameters.	- Most parameters are clear. - Parameters require some fine tuning to obtain good results.	- Parameters are clear - Parameters required minimal fine tuning obtain good results.	Note which parameters are unclear or required a lot of tweaking.
Simplicity of process	Unclear how to proceed with executing the tool	After some trial and error, process completed successfully	Straightforward operation	Note any suggestions for improving the workflow.
GUI (when applicable)	- Issues with functionality of GUI and selecting options - Order of buttons in GUI doesn't match workflow	A few options\elements of GUI are unclear but generally easy to follow.	GUI is intuitive and easy to run.	Note any confusing elements of the GUI.
User Manual	User manual missing key steps.	User manual explains key steps but portions are confusing.	User manual explains tool well, easy to follow	Point out specific things to add to the user manual or portions that are confusing.

Future Explorations

- What opportunities are there for future GIS-based tools to improve cleanup of the dataset? e.g. GIS scripts to combine lines or segment as were developed for SPR799.
- What is needed to transition this tool to work with the new Riegl scanner? (e.g. trajectory information)?

APPENDIX E: FEATURE EXTRACTION SUMMARY

Table E-1. Summary of features in all data files considered for training data.

Labeled Data ID	File ID	Crosswalk			Median			Bike lane			Lane Marking	
		Std. Parallel	Stag. Cont.	Ladder	Raised	Closed	Painted	Bike Symbol	Std. Bike Lane	Buff. Bike Lane	White Solid	White Dash
SiteID_1_Hwy99EB												
	1_HWY99EB_00										1	-
	1_HWY99EB_01	1						1			1	-
	1_HWY99EB_02	1						1	1		1	-
	1_HWY99EB_03	1						1			1	-
	1_HWY99EB_04							1			1	-
	1_HWY99EB_05	1						1			1	-
	1_HWY99EB_06							1			1	-
4	1_HWY99EB_07	1						1			1	-
	1_HWY99EB_08										1	-
	1_HWY99EB_09	1						1			1	-
	1_HWY99EB_10	1						1			1	-
	1_HWY99EB_11	1									1	-
	1_HWY99EB_12	1						1			1	-
5	1_HWY99EB_13	1						1			1	-
	1_HWY99EB_14							1			1	-
	1_HWY99EB_15	1						1			1	-
	1_HWY99EB_16	1			1			1			1	-
	1_HWY99EB_17	1						1			1	-
	1_HWY99EB_18							1			1	-
6	1_HWY99EB_19			1				1			1	-

Labeled Data ID	File ID	Crosswalk			Median			Bike lane			Lane Marking	
		Std. Parallel	Stag. Cont.	Ladder	Raised	Closed	Painted	Bike Symbol	Std. Bike Lane	Buff. Bike Lane	White Solid	White Dash
7	1_HWY99EB_20	1						1			1	-
	1_HWY99EB_21	1									1	-
	1_HWY99EB_22	1						1			1	-
	1_HWY99EB_23							1			1	-
8	1_HWY99EB_24	1						1			1	-
	1_HWY99EB_25	1									1	-
	1_HWY99EB_26	1									1	-
	1_HWY99EB_27										1	-
	1_HWY99EB_28										1	-
	1_HWY99EB_29										1	-
SiteID_1 Hwy221												
	1_HWY221_01				1						1	
	1_HWY221_04			1	1						1	1
13	1_HWY221_06	1		1	1			1			1	
	1_HWY221_07	1		1	1			1			1	
	1_HWY221_10										1	
	1_HWY221_13										1	
	1_HWY221_02										1	
	1_HWY221_05										1	
24	1_HWY221_08	1		1	1			1			1	1
25	1_HWY221_10	1		1	1			1			1	
	1_HWY221_11	1			1			1			1	1
26	1_HWY221_12			1	1			1			1	1
	1_HWY221_14				1			1			1	1
27	1_HWY221_17	1			1			1			1	
SiteID_2 Hwy99W												
	2_HW99WSB_21	1						1			1	

Labeled Data ID	File ID	Crosswalk			Median			Bike lane			Lane Marking	
		Std. Parallel	Stag. Cont.	Ladder	Raised	Closed	Painted	Bike Symbol	Std. Bike Lane	Buff. Bike Lane	White Solid	White Dash
	2_HW99WSB_24	1						1			1	1
	2_HW99WSB_26	1						1			1	1
	2_HW99WSB_28										1	1
	2_HW99WSB_30	1						1			1	1
	2_HW99WSB_32							1			1	1
23	2_HW99WSB_34	1	1		?			1			1	1
	2_HW99WSB_36	1						1			1	1
	2_HW99WSB_38	1						1			1	1
	2_HW99WSB_40							1			1	1
28	2_HW99WNB_18	1	1		1			1			1	1
29	2_HW99WNB_21							1			1	1
	2_HW99WNB_24	1						1	1		1	
	2_HW99WNB_26	1									1	
	2_HW99WNB_28	1	1		1			1			1	1
	2_HW99WNB_30	1									1	
	2_HW99WNB_32	1						1			1	
	2_HW99WNB_34	1						1			1	1
30	2_HW99WNB_36				1		1	1			1	
	2_HW99WNB_38	1						1			1	
SiteID_2 Hwy141												
	2_HW141_6	1						1			1	
	2_HW141_8	1						1	1		1	1
	2_HW141_10	1							1		1	1
	2_HW141_12	1	1					1				
20	2_HW141_14	1	1				1	1			1	1
	2_HW141_16	1						1			1	
	2_HW141_18							1			1	

Labeled Data ID	File ID	Crosswalk			Median			Bike lane			Lane Marking	
		Std. Parallel	Stag. Cont.	Ladder	Raised	Closed	Painted	Bike Symbol	Std. Bike Lane	Buff. Bike Lane	White Solid	White Dash
21	2_HW141_20	1						1	1		1	1
22	2_HW141_22	1						1			1	1
SiteID_3 Hwy29												
	3_HW29_5	1	1		1			1			1	1
	3_HW29_11	1			1			1			1	
	3_HW29_17	1									1	
	3_HW29_23	1						1			1	
	3_HW29_29							1			1	1
14	3_HW29_35	1						1	1		1	1
	3_HW29_41	1									1	1
	3_HW29_47										1	
	3_HW29_53	1									1	
15	3_HW29_59	1			1			1			1	1
	3_HW29_65										1	
	3_HW29_1										1	
	3_HW29_8	1										
	3_HW29_15										1	
	3_HW29_22	1						1			1	1
	3_HW29_29										1	
	3_HW29_36				1			1			1	1
	3_HW29_43							1			1	1
	3_HW29_50	1									1	
	3_HW29_57	1						1			1	
	3_HW29_64	1						1			1	
SiteID_3 Hwy102												
9	3_HW102_0	1	1					1			1	1
10	3_HW102_1							1			1	

Labeled Data ID	File ID	Crosswalk			Median			Bike lane			Lane Marking	
		Std. Parallel	Stag. Cont.	Ladder	Raised	Closed	Painted	Bike Symbol	Std. Bike Lane	Buff. Bike Lane	White Solid	White Dash
	3_HW102_2							1			1	
	3_HW102_3							1			1	
11	3_HW102_4	1						1			1	1
12	3_HW102_5		1		1							
	3_HW102_6				1			1			1	
SiteID_4 Hwy99E												
	4_HW99E_25	1									1	
	4_HW99E_28	1									1	
	4_HW99E_31										1	
	4_HW99E_34	1									1	
	4_HW99E_36	1									1	
	4_HW99E_38	1						1			1	1
	4_HW99E_40							1			1	
	4_HW99E_42	1					1	1			1	1
31	4_HW99E_44	1						1	1		1	1
32	4_HW99E_46	1						1	1		1	1
	4_HW99E_0										1	
	4_HW99E_2	1						1	1		1	1
	4_HW99E_4	1						1	1		1	1
	4_HW99E_6	1					1	1			1	
	4_HW99E_8	1									1	
33	4_HW99E_10		1					1			1	
	4_HW99E_12	1						1			1	
	4_HW99E_14	1									1	
	4_HW99E_16											
	4_HW99E_18											
	4_HW99E_20										1	

Labeled Data ID	File ID	Crosswalk			Median			Bike lane			Lane Marking	
		Std. Parallel	Stag. Cont.	Ladder	Raised	Closed	Painted	Bike Symbol	Std. Bike Lane	Buff. Bike Lane	White Solid	White Dash
	4_HW99E_22							1			1	
	4_HW99E_24	1						1			1	
SiteID_5 Hwy213												
	5_HW213SB_1				1						1	
	5_HW213SB_2						1				1	
	5_HW213SB_4	1										
	5_HW213SB_6	1										
	5_HW213SB_8	1										
	5_HW213SB_10						1					
	5_HW213SB_12	1										
	5_HW213SB_14	1						1	1		1	
1	5_HW213SB_18	1					1					
	5_HW213SB_17	1	1		1		1	1	1		1	
	5_HW213SB_20	1										
	5_HW213SB_22	1										
	5_HW213SB_24	1										
2	5_HW213SB_26	1	1					1				1
	5_HW213SB_28						1					
3	5_HW213SB_30		1		1							
16	5_HW213NB_8	1						1	1			1
	5_HW213NB_10	1										
	5_HW213NB_12	1					1					
	5_HW213NB_14	1						1	1			
	5_HW213NB_16	1										
	5_HW213NB_18	1										
	5_HW213NB_20	1										
	5_HW213NB_22	1					1					

Labeled Data ID	File ID	Crosswalk			Median			Bike lane			Lane Marking	
		Std. Parallel	Stag. Cont.	Ladder	Raised	Closed	Painted	Bike Symbol	Std. Bike Lane	Buff. Bike Lane	White Solid	White Dash
17	5_HW213NB_24	1			1		1	1				
18	5_HW213NB_26		1	1								
	5_HW213NB_27				1		1					
19	5_HW213NB_28	1		1	1						1	
	5_HW213NB_30	1										
	5_HW213NB_32	1										
	5_HW213NB_34	1					1					
	5_HW213NB_36										1	
	5_HW213NB_38	1			1			1			1	
Other Data												
	Other	1	1		1			1				
Totals (All Files)		104	14	9	28	0	14	91	14	0	126	41
Total (Labeled Data)		24	9	7	12	0	4	28	5	0	26	16