STATE OF CALIFORNIA • DEPARTMENT OF TRANSPORTATION

TECHNICAL REPORT DOCUMENTATION PAGE

TR-0003 (REV 04/2024)

1. REPORT NUMBER	2. GOVERNMENT ASSOCIATION NUMBER	3. RECIPIENT'S CATALOG NUMBER
CA25-4189	N/A	N/A

4. TITLE AND SUBTITLE	5. REPORT DATE
Point Cloud Feature Extraction for ADA Ramp Compliance Assessment	04/28/2025
	6. PERFORMING ORGANIZATION CODE
	AHMCT Research Center, UC Davis
7. AUTHOR	8. PERFORMING ORGANIZATION REPORT NO.
Amin Ghafourian, Andrew Lee, Dechen Gao, Tyler Beer, Kin Yen, Iman Soltani	4189 - UCD-ARR-25-01-31-01
9. PERFORMING ORGANIZATION NAME AND ADDRESS AHMCT Research Center UCD Dept. of Mechanical & Aerospace Engineering Davis, California 95616-5294	10. WORK UNIT NUMBER N/A
	11. CONTRACT OR GRANT NUMBER 65A0749, Task 4189
12. SPONSORING AGENCY AND ADDRESS California Department of Transportation P.O. Box 942873, MS #83 Sacramento, CA 94273-0001	13. TYPE OF REPORT AND PERIOD COVERED Final Report, 11/2022-1/2025
	14. SPONSORING AGENCY CODE Caltrans

15. SUPPLEMENTARY NOTES

N/A

16. ABSTRACT

Automation can play a prominent role in improving efficiency, accuracy, and scalability in infrastructure surveying and assessing construction and compliance standards. This report presents a comprehensive framework for automation of geometric measurements and compliance assessment using point cloud data. The proposed approach integrates deep learning-based detection and segmentation with classical geometric and signal processing techniques to automate surveying tasks. The framework is applied to assess Americans with Disabilities Act (ADA) compliance of curb ramps, demonstrating the utility of point cloud (PC) data for automatic ramp extraction, segmentation, and calculation of geometric measurements. The method leverages a new large annotated dataset of ramps, facilitating robust deep model training and evaluation. Experimental results, including manual field-based and PC-based measurements of several ramps, validate the accuracy and reliability of the proposed method across various scenarios, highlighting its potential to significantly reduce manual effort and improve consistency in infrastructure assessment. Beyond ADA compliance, the proposed framework establishes a foundation for broader applications in surveying and automated evaluation of construction infrastructure, paving the way for more widespread adoption of point cloud data in engineering automation.

17. KEY WORDS	18. DISTRIBUTION STATEMENT
Automated Survey, ADA compliance assessment, MTLS, LiDAR, Mobile Terrestrial Laser Scanning	No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.

19. SECURITY CLASSIFICATION *(of this report)*	20. NUMBER OF PAGES	21. COST OF REPORT CHARGED
Unclassified	48	N/A

DISCLAIMER

This document is disseminated in the interest of information exchange. The contents of this report reflect the views of the authors who are responsible for the facts and accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of California or the Federal Highway Administration. This publication does not constitute a standard, specification or regulation. This report does not constitute an endorsement by the Department of any product described herein.

For individuals with sensory disabilities, this document is available in alternate formats. For information, call (916) 654-8899, TTY 711, or write to California Department of Transportation, Division of Research, Innovation and System Information, MS-83, P.O. Box 942873, Sacramento, CA 94273-0001.

Advanced Highway Maintenance and Construction Technology Research Center

Department of Mechanical and Aerospace Engineering
University of California at Davis

Point Cloud Feature Extraction for ADA Ramp Compliance Assessment

Amin Ghafourian, Andrew Lee, Dechen Gao, Tyler Beer, Kin Yen & Iman Soltani: Principal Investigator

Report Number: CA25-4189

AHMCT Research Report: 4189 - UCD-ARR-25-01-31-01

Final Report of Contracts: 65A0749, Task 4189

April 28, 2025

California Department of Transportation

Division of Research, Innovation and System Information

Executive Summary

Problem, Need, and Purpose of Research

Manual surveying of infrastructure assets remains a costly and labor-intensive process despite the availability of rich digital datasets such as mobile LiDAR scans. This burden is especially evident in compliance assessments like those required by the Americans with Disabilities Act (ADA), where thousands of curb ramps must be precisely measured. The purpose of this research is to develop and validate an automated framework that combines modern machine learning techniques with geometric modeling to automate surveying tasks, improve consistency, reduce labor, and enhance scalability for infrastructure management.

Overview of Work and Methodology

This work proposes a hybrid automation framework that integrates deep learning-based asset detection and segmentation with classical geometric and signal processing methods to perform high-accuracy measurements from point cloud data to check curb ramp ADA compliance. The methodology includes ramp detection through 2D projection and object detection models, detailed component segmentation using an adapted image segmentation model, geometric refinement using design priors and geometric primitives, reference point extraction facilitated through the proposed score-based line fitting, and measurement acquisition on point cloud data given geometric reference points. A human-annotated dataset of curb ramps, along with manual field measurements, supports training, evaluation, and validation of the proposed system.

Major Results and Recommendations

The automated system successfully detects and processes curb ramps with high reliability, achieving a measurement agreement rate of nearly 88% with manual field inspections, rising to over 97% agreement when small tolerance margins are allowed. The system demonstrates particularly strong performance on slope-related measurements, while minor discrepancies in width measurements are mainly attributed to surface irregularities and visual ambiguity in field surveys. Quality control steps are essential to ensure reliability, particularly in filtering ramps with poor data density or severe geometric anomalies.

We recommend that Caltrans adopt quality control procedures similar to, and expanding upon, those outlined in this study, using digital point cloud quality (such as density thresholds) as an early filter to ensure reliable automated surveying outcomes. Although nearly half of the ramps in our evaluation dataset were disqualified due to low data quality or severe geometric

deviations, these disqualifications correctly flagged assets that were either unsuitable for automation or already severely non-compliant. Thus, quality control itself can serve as an early warning mechanism. We also recommend that future mobile LiDAR data collections by Caltrans be designed with automation in mind, ensuring sufficient density and completeness to reduce disqualification rates and maximize automation efficiency. Finally, while the current pipeline already enables significant labor reduction by automatically processing nearly half of the ramps, future work should focus on enabling a similar analysis on other design types and assets, as well as improving performance on irregular designs and occluded assets, and leveraging multimodal data sources to further enhance robustness.

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

Acronym	Definition
AHMCT	Advanced Highway Maintenance and Construction Technology Research Center
ADA	Americans with Disabilities Act
AI	Artificial Intelligence
Caltrans	California Department of Transportation
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DGCNN	Dynamic Graph Convolutional Neural Network
DL	Deep Learning
DOTs	Departments of Transportation
DRISI	Caltrans Division of Research, Innovation and System Information
iForest	Isolation Forest
LiDAR	Light Detection And Ranging
ML	Machine Learning
MMCN	Multi-faceted Multi-object Convolutional Neural Network
MTLS	Mobile Terrestrial Laser Scanning
OCSVM	One-Class Support Vector Machines
PC	Point Cloud
PCA	Principal Component Analysis
SAM	Segment Anything Model
SBLF	Score-Based Line Fitting

Acronym	Definition
STLS	Stationary Terrestrial Laser Scanning
SVM	Support Vector Machine
UCD	University of California – Davis

Acknowledgments

The authors thank the California Department of Transportation (Caltrans) for their support, in particular Aaron Chamberlin and Kenneth Sutterfield with the Division of Construction, and Jessaneil Perez and Mohey El-Mously with the Division of Research, Innovation and System Information. The authors thank Districts 1, 2, 3, 4, 6, 11, and 12 surveyors for their contributions of ADA ramp point cloud data. The authors acknowledge the dedicated efforts of the AHMCT team and project panel members who have made this work possible.

Chapter 1: Introduction

Background and Problem

Surveying plays a pivotal role in city management and the maintenance of urban infrastructure, ensuring the functionality, safety, and accessibility of public spaces. Accurate and timely surveys are essential for assessing the condition of infrastructure, identifying signs of deterioration, and planning necessary repairs or upgrades. These activities also serve a critical function in verifying compliance with design specifications, construction standards, and regulatory frameworks. For instance, adherence to the Americans with Disabilities Act (ADA) requirements ensures equitable access to public infrastructure for all individuals. By systematically documenting infrastructure conditions and compliance, surveying supports effective resource allocation and minimizes potential liabilities associated with non-compliance or unsafe conditions. As cities expand and age, the demand for efficient and accurate surveying continues to grow.

Despite its critical importance, surveying remains an exceptionally labor-intensive process due to the sheer scale and diversity of infrastructure and building assets that require regular assessment. Urban environments encompass a vast array of features, including roadways, sidewalks, bridges, parking facilities, drainage systems, and utilities, each with their own unique set of compliance requirements and measurement criteria. For example, ensuring ADA compliance involves assessing thousands of curb ramps at pedestrian crossings and measuring parameters such as ramp slope, cross slope, width, etc. Similarly, evaluating the structural integrity of bridges demands detailed inspections of girders, bearings, and joints, while roadway surveys may require precise measurement of pavement conditions, lane markings, and traffic signage. The extensive variety and geographic distribution of these assets necessitate significant manpower and time, often leading to surveying delays and increased costs. As urban areas continue to grow, the burden of manual surveying is becoming increasingly unsustainable, emphasizing the urgent need for automation.

In recent years, the California Department of Transportation (Caltrans) has increasingly turned to mobile terrestrial scanning technologies to document and manage infrastructure more effectively. These systems generate high-resolution, georeferenced point cloud data and registered images, providing a detailed

and comprehensive digital representation of civil infrastructure. These point clouds are commonly referred to as digital twins. Digital twins serve as invaluable resources for infrastructure management, offering unprecedented opportunities for monitoring, planning, and decision-making. However, despite the growing availability of such rich datasets, their potential remains largely untapped for automating critical surveying tasks. Current efforts have predominantly focused on visualization and manual data interpretation, leaving a significant gap in leveraging these resources to streamline and enhance infrastructure assessment processes. By developing automated methods to analyze and extract actionable information from these data, the full utility of digital twins can be realized, unlocking efficiencies that are yet to be achieved in surveying.

Recent advancements in artificial intelligence (AI), particularly in machine learning (ML) and deep learning (DL), present transformative opportunities for automating surveying tasks, offering the potential to save millions of dollars annually. These technologies excel in tasks such as feature/object detection, segmentation, and classification, making them valuable tools for processing large-scale data efficiently. However, sole reliance on end-to-end AI-based solutions is not sufficient for the quantitative demands of surveying in which precision and accuracy are paramount. Such end-to-end methods often require vast amounts of labeled training data, which are time-consuming to generate and may still fall short of achieving the stringent accuracy needed for many surveying applications. There is a need for methods that automate the survey process using the available digital data.

Objectives and Contributions

The objective of this work is to develop reliable automated tools for survey measurements using the available terrestrial scanning data. We propose a hybrid methodology that integrates the strengths of modern AI tools with conventional geometric analysis techniques. This approach leverages the robust capabilities of AI for detection and segmentation while minimizing reliance on their precision by complementing them with classical geometric analysis methods for accurate quantitative measurements. By combining the scalability of AI with the reliability of geometric analysis, we aim to achieve a generalizable and scalable framework for automating infrastructure surveying tasks across diverse applications.

We validate this approach through the specific application of assessing curb ramps at pedestrian crossings. This task exemplifies the broader challenges of surveying, as ensuring ADA compliance requires precise geometric measurements to guarantee accessibility and safety. Accurate assessment of curb ramps requires the measurement of numerous geometric features to ensure compliance with ADA standards, including the slope, cross slope, width,

and surface deviations of the main ramp; the area featuring truncated domes; the flares; and gutter and landing areas (Appendix A).

Such measurements are often prone to bias and inaccuracy due to the reliance on manual labor as workers frequently use aesthetic visual cues—such as concrete lines or surface patterns—to delineate ramp components. These cues may not correspond to the actual boundaries required for precise measurement, leading to inconsistent results. Compounding the challenge, is the fact that many ramps fail to meet ADA requirements during initial construction due to errors. Over time, other factors, such as soil settling, erosion, or general wear and tear, can further degrade compliance, turning once-compliant ramps into liabilities. These various factors necessitate repeated measurements over the lifecycle of the infrastructure, imposing significant labor and costs while still falling short of the reliability and precision that modern urban management demands. These issues, in one way or another, are shared across various surveying tasks. For example, assessing roadway pavement conditions involves measuring surface smoothness, crack depths, and wear patterns, tasks that similarly rely on subjective visual assessments and are prone to error. Given these challenges, and bearing in mind the sheer quantity of such assets in urban environments, underscores the broader need for scalable, automated solutions that can deliver the accuracy and consistency necessary for infrastructure surveying while reducing dependence on labor-intensive and costly manual methods.

Focusing on automating ADA compliance assessment for curb ramps at pedestrian crossings, this work provides a publicly available dataset containing nearly 1,500 annotated and segmented ramps, representing one of the largest datasets of its kind. The framework implements a novel hybrid approach to extract quantitative measurements. While the methodologies are developed specifically for ADA ramps, they are designed to be adaptable to other surveying tasks. Additionally, this work includes field-measured data as well as digital point cloud measurements performed by experts and trained personnel, ensuring a reliable benchmark for validation. By combining shared datasets, ground-truth measurements, and open-source codes and algorithms, this effort paves the way for researchers to further advance methodologies and evaluate their approaches against validated manual measurements.

Prior Work

Recently, there have been some research efforts that leverage the available digital data, such as point clouds, and may indirectly benefit automation of surveying efforts. A notable study [1] introduced a DL-based approach for point cloud classification using transformers, aiming to improve the accuracy of classified point clouds for applications such as urban planning and infrastructure management. In the realm of bridge inspection, researchers [2] compared

three DL models—PointNet, PointCNN, and Dynamic Graph Convolutional Neural Network (DGCNN)—for classifying bridge components from point cloud data, highlighting the potential of DL in automating structural inspections. Additionally, other researchers [3] proposed a multi-faceted multi-object convolutional neural network (MMCN) combined with a support vector machine (SVM) for the fully automated classification of highly dense 3D point cloud data acquired from mobile LiDAR systems, demonstrating applications in high-density maps and highway monitoring. Despite these advancements, fully automating a broader range of surveying tasks remains an emerging field, indicating significant opportunities for future research and application.

Chapter 2:
Data-Driven, Hybrid ML/Analytical Framework for Survey Automation

Survey automation in infrastructure management requires a structured methodology that combines modern ML techniques with classical analytical tools to achieve both scalability and precision. The proposed framework begins with identifying assets of interest in MTLS data, leveraging advanced object detection techniques. Once the asset is located, segmentation methods are applied to isolate its specific components, either in the raw 3D point cloud or through 2D projections or using associated RGB images. Following segmentation, analytical techniques are employed to decompose the asset into its fundamental geometric primitives, such as planes, lines, and points. These primitives enable the determination of critical geometric reference points, including intersections and boundaries, which serve as anchors for making quantitative measurements. By integrating ML-based data processing with analytical methodologies, this hybrid framework ensures accurate, repeatable, and scalable automation of surveying tasks, addressing the limitations of manual labor and end-to-end ML approaches.

Asset Detection and Extraction

The first step in the proposed framework is the identification of assets of interest within the MTLS data. This process involves applying ML-based object detection techniques, leveraging either 2D image data, 2D projections of 3D point cloud, raw 3D point clouds, or a combination of these approaches. For 2D approaches, state-of-the-art object detection models, such as YOLOv8 [4] and Faster R-CNN [5], have demonstrated robust performance in identifying infrastructure elements from registered images. These methods can also be applied to 2D projections of point clouds [6] [7] [8], such as bird's eye views or height maps, offering a simpler, yet effective, way to leverage 2D object detection tools for 3D data. For 3D object detection, models like PointRCNN [9] and VoteNet [10] have emerged as leading solutions specifically designed to operate directly on point cloud data by identifying objects while preserving their spatial properties. Hybrid approaches that combine image and point cloud data, such as Multi-View CNNs or multimodal fusion networks [11], further enhance detection accuracy by integrating complementary features from both data modalities. This step is crucial as the accurate localization of the asset forms the foundation for subsequent processes, ensuring that only relevant portions of the digital terrestrial scanning dataset are analyzed in later stages.

Segmentation

Once the asset of interest is identified, the next step involves segmenting it into its constituent parts to enable detailed analysis and measurement. Similar to the detection stage, segmentation can be approached using either 2D or 3D techniques, depending on the data modality and application requirements. For 2D data, semantic segmentation models like DeepLabV3+ [12], U-Net [13] [14] and more recently, the Segment Anything Model (SAM) [15] methods have proven highly effective, offering pixel-level classification in registered images or 2D projections of point clouds. On the other hand, 3D segmentation techniques, such as those based on PointNet++ [16], Kernel Point Convolution (KPConv) [17], or RandLA-Net [18], directly operate on raw point cloud data, preserving spatial geometry while assigning labels to individual points. These methods excel in parsing complex, unstructured datasets to isolate meaningful regions. Recent advancements also highlight the use of hybrid segmentation approaches whereby 2D projections are segmented first, and the results are mapped back to the 3D domain to enhance computational efficiency and accuracy [19] [20] [21].

Geometric Primitives

The third step in the proposed framework focuses on extracting geometric primitives, leveraging the inherent assumption that man-made structures are often composed of simple, well-defined geometric elements, such as planes, edges, and lines. The segmentation results from the previous stage serve as input, providing groups of points that can be fitted to these geometric primitives. Simple techniques, such as plane fitting, edge detection, and geometric modeling, are employed in this stage to identify the most reliable primitives while filtering out noise and outliers in the data. By rooting this process in the known geometric structure of the asset, the extracted primitives provide a more accurate representation than segmentation results alone, ensuring that the building blocks of the asset are precisely defined. This approach is particularly critical for addressing the imperfections and noise inherent in MTLS data or segmentation outputs. Untreated noise can significantly impact the accuracy of quantitative measurements, but extracting primitives based on a geometric model mitigates these errors, enabling a cleaner and more structured representation of the assets of interest. This stage lays a foundation for the identification of geometric reference points in the next step.

Survey Reference Points

The final step in the proposed framework involves extracting measurement reference points, which play a critical role in obtaining key quantitative

measurements, such as slopes, widths, surface deviations, and other essential parameters. Building on the geometric primitives extracted in the previous step, intersections of planes are identified to form lines, and intersections of lines are further refined to yield points that represent the boundaries of the asset in a minimalistic and abstract form. These reference points serve as the foundation for precise measurements, allowing the framework to quantify various aspects of the asset with accuracy. To ensure reliability, this and the previous step often require iterative refinements. The extracted reference points are tested against existing models and domain-specific knowledge of the asset, such as realistic ranges of distances, angles, or slopes. Extreme and unrealistic discrepancies in these tests may highlight the need to revisit the geometric primitives extraction stage, refining and repeating the process to improve the quality of the reference points. This iterative approach ensures that the reference points are robust and aligned with the structural and functional model of the asset. Once reference points are established, they can be used to derive final measurements. Figure 2.1 presents a flowchart of the proposed framework.

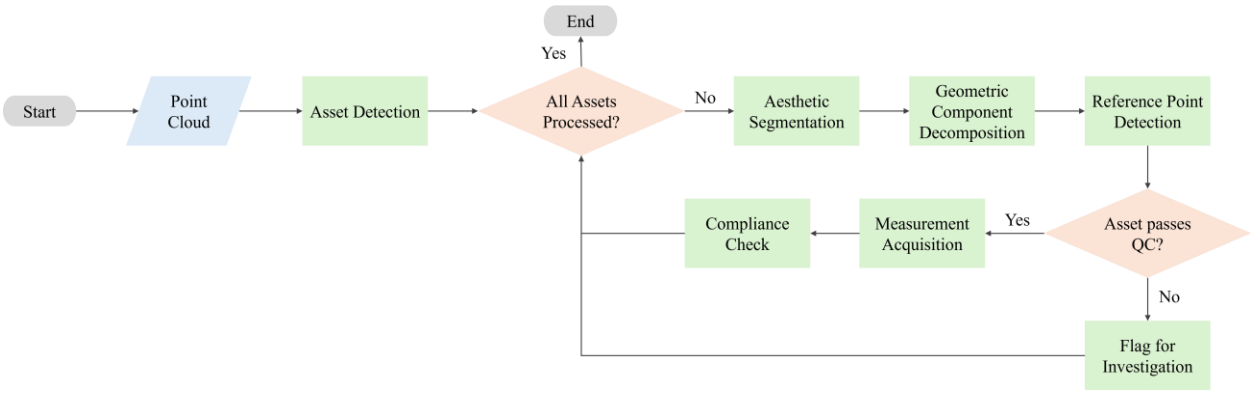

Figure 2.1: Flowchart of the proposed ADA ramp compliance assessment automation method based on point cloud data.

Chapter 3:
Automated ADA Ramp Compliance Assessment

Background and Need

In recent years, the State of California has faced significant legal challenges regarding the non-compliance of its infrastructure with ADA regulations. In 2009, the Caltrans settled a lawsuit by pledging $1.1 billion to improve sidewalk access statewide, addressing deficiencies such as the absence of curb ramps and detectable warnings [22]. Similarly, in 2015, the City of Los Angeles resolved a landmark class-action lawsuit by committing $1.4 billion over 30 years to repair sidewalks and install curb ramps, ensuring compliance with ADA requirements [23] [24]. These settlements represent some of the largest disability access-related agreements in U.S. history, highlighting the critical need for rigorous compliance with ADA standards to ensure accessibility and safety for all users.

Current Manual Procedure for ADA Compliance Assessment

In the conventional manual approach, ramp measurements are conducted manually in the field by trained personnel. Depending on the ramp type, specific parameters, such as slopes, cross slopes, widths, surface deviations, and dimensions of features like truncated domes, flares, gutter and landing areas, must be recorded and compared against approved reference values. For this study, we focus on four primary ramp types, referred to as Types A through D, each requiring a unique set of measurements based on their geometric characteristics. For each ramp, field personnel are required to take approximately 45 distinct measurements. These measurements are typically performed using tools such as inclinometers and measuring tapes, often relying on visual cues to delineate boundaries. Once the measurements are collected, the data are populated into a standardized PDF form (Appendix A), which serves as an official record for compliance assessment. This method, while thorough, is highly labor-intensive, prone to errors, and limited by the subjective interpretation of field personnel, particularly when clear geometric boundaries are absent.

Appendix A illustrates the types of measurements required for ramp Types A through D, highlighting the diversity and complexity of the manual assessment process.

Dataset

In recent years, Departments of Transportation (DOTs) across the United States, including Caltrans, have increasingly adopted advanced technologies such as static and MTLS to create detailed digital representations of urban infrastructure. This shift towards digitalization has been driven by the need for accurate, efficient, and safe methods of managing transportation assets. Caltrans, for instance, owns and operates six MTLS systems using Trimble MX9 and MX50 systems. These systems have been instrumental in collecting high-precision survey-grade point cloud data. Caltrans has attempted to replace manual field measurement with manual measurement on digital point cloud data. The integration of MTLS technology into Caltrans' workflow has led to increased employee safety, reduced costs, and expedited project delivery [25]. The accumulation of extensive point cloud datasets presents significant opportunities for cost savings and operational efficiency. By leveraging these data, DOTs can automate asset management, monitor infrastructure health, and ensure compliance with standards such as the ADA. This reflects a broader trend towards digital transformation in infrastructure management.

In this work, we utilize extensive digital data collected by Caltrans from cities, such as Woodland, Sacramento, and Chico, etc., which have been captured using MTLS systems. While these datasets provide a comprehensive representation of urban infrastructure, they lack the annotations necessary for tasks such as ramp detection and segmentation. To address this gap, we manually annotated a substantial portion of the data, focusing on ramps and their key components. Figure 3.1 illustrates examples of segmented ramps from the dataset. The annotated dataset, which we make publicly available, includes nearly 1700 detected and segmented ramps. By sharing this resource, we aim to enable other research groups to build upon this work, advancing the automation of infrastructure management and surveying using existing digital data.

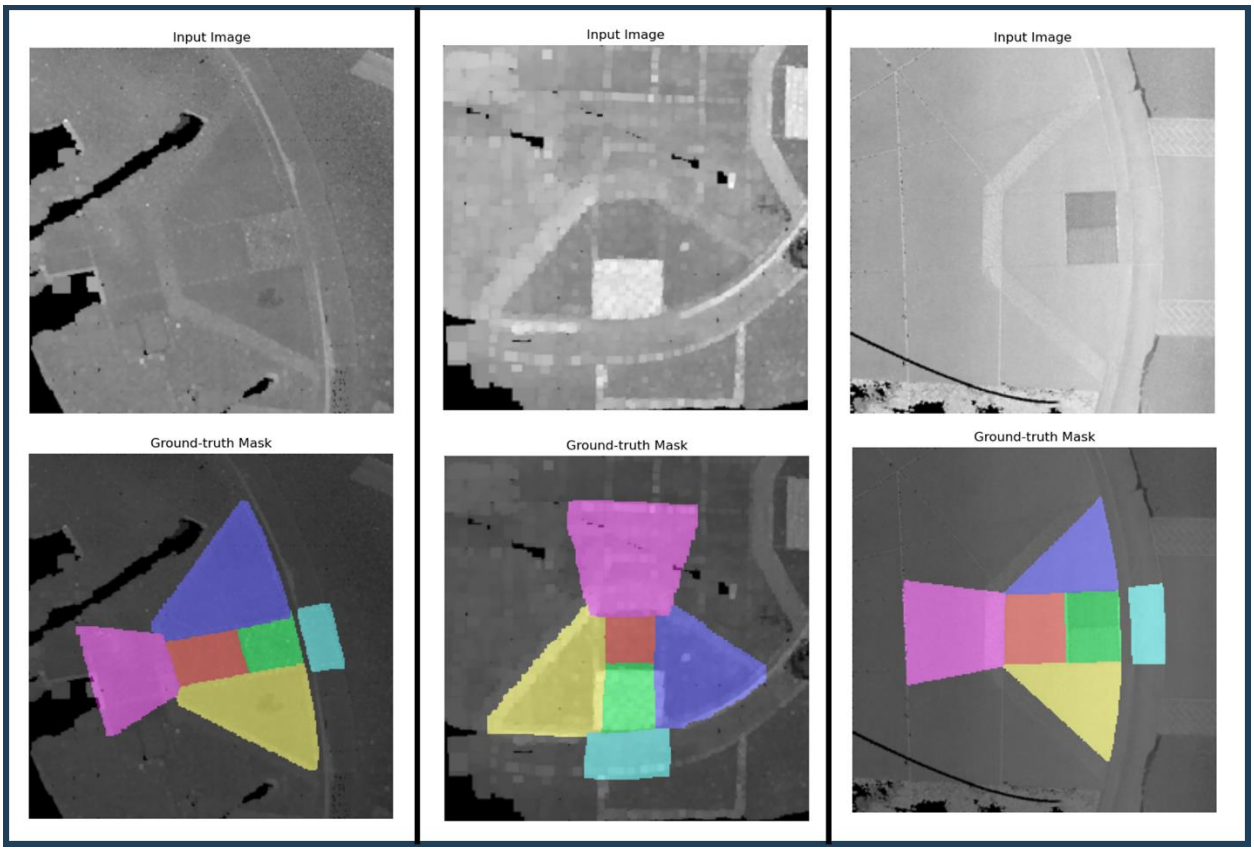

Figure 3.1: Example annotations indicating various components of an ADA ramp.

Ramp Detection

This section illustrates the ramp detection method, which aims to localize and extract individual ramps from a large point cloud with high accuracy and reliability for downstream processing and measurement extraction.

Comparison of 3D and 2D Object Detection

For ramp detection, both 3D and 2D object detection techniques were considered. 3D object detection directly processes the point cloud data, leveraging the full spatial structure of the environment. While this method generally captures detailed geometric information, it usually requires significantly higher computational resources and complex models, and subsequently more training data, making it less efficient. In addition, ramps often have minimal distinct features from the adjacent sidewalks, blending with them in terms of color and surface texture. This lack of clear boundaries makes it challenging for 3D object detection models to reliably isolate ramps from their surroundings.

On the other hand, 2D object detection simplifies the problem by projecting the 3D point cloud into a 2D representation. In 2D, we can make use of well-

established and robust 2D object detection models and reduce computational complexity. Although some spatial information is lost in the top-down projection, the normalized grayscale intensity values compensate for such losses by providing a consistent representation of edges, textures, and surface variations. This factor enables the detection model to distinguish ramps from surrounding structures effectively. As such, we chose 2D object detection methods for initial data processing and extracting the ADA ramps from the point cloud.

Point Cloud Preprocessing

We compared Faster R-CNN [5] and DeTR [26] for ramp detection, and chose DeTR as our final object detection model. Faster R-CNN is a well-established 2D object detection model with a two-stage approach. While Faster R-CNN is effective for general object detection, in our application it struggled with small or indistinct features, which are critical for isolating ramps from the surroundings. DeTR employs a transformer-based architecture that predicts object positions and classes in an end-to-end manner. It excels at capturing contextual information throughout the entire image by leveraging a global attention mechanism, which makes it particularly effective for detecting ramps with minimal distinctions from their surroundings.

For both models, we fine-tuned pre-trained weights on our ramp detection dataset. Since inference speed is not a primary concern in this scenario, we evaluated both ResNet-50 and ResNet-101 backbones. We then selected the ResNet-101 as the backbone for our DeTR model due to its superior performance.

Data Preprocessing for Inference

During deployment on unseen data, we follow the same preprocessing steps used during training. The point cloud is divided into overlapping square patches and projected top-down onto a 1280 by1280 canvas. Intensity values are normalized to grayscale, and the transformation metadata necessary for mapping predictions back to 3D space is preserved. These preprocessed 2D images are then fed into the trained ramp detection model, which outputs 2D bounding boxes for the ramp class.

Extracting Individual Ramps

As a result of ramp detection, bounding boxes are predicted for each 2D image. Using the transformation metadata, the 2D image and the corresponding bounding box coordinates are accurately converted back to 3D space. The predicted 3D bounding boxes are then mapped back to the original point cloud. Each bounding box is extended by moving its edges outwards by 10% of their original length to include any adjacent points that may belong to the ramp but fall outside the detected boundary. Finally, the

extracted ramps are saved in separate files for downstream segmentation and measurement.

Ramp Segmentation based on Visual Cues

This section presents our approach for ramp segmentation based on visual cues. The segmentation is challenged by (a) data scarcity; (b) the high dimensionality of 3D point clouds, which complicates segmentation; and (c) non-uniform point density, requiring robustness to point clouds of varying quality and density.

In order to address (a) and (b), we carry out the segmentation in the 2D image space using the SAM [15], a strong pretrained image segmentation model that can efficiently adapt to custom segmentation tasks with minimal samples. To convert point clouds to 2D images and mitigate the effect of density variation as specified in (c), we incorporate additional steps for preprocessing, which we describe in the following section.

Point Cloud Preprocessing

To visualize the point cloud as a 2D image, the projection of the 3D point cloud onto the horizontal plane is visualized as a grayscale image where different shades of gray reflect different LiDAR intensity values. This preprocessing, however, outputs a sparsely colored image with varying local densities, reflecting the local PC density, resulting in poor segmentation performance. As such, we apply pixel dilation to fill out the empty areas and form contiguous regions.

Dilation, however, faces additional challenges. Notably, varying point cloud densities prevent a one-size-fits-all dilation approach. If we consider a small kernel, it will miss significant empty areas, leaving the lower-density areas still sparse and discontinuous. On the other hand, a large kernel can obscure detailed ramp information and most importantly, ramp separation boundaries.

To address these challenges, we propose an adaptive dilation approach, which tailors the dilation to local density with variable kernel sizes, vastly improving contiguity while preserving detail. The process involves chunking the ramp images and computing the local density for each chunk and calculating and applying a corresponding optimal kernel size for each one. The square dilation kernel size κ for the chunk is calculated by the following formula:

$$\kappa = \min\left(\left\lfloor \frac{1}{2\rho_{proj}} \right\rfloor, \kappa_{max}\right)$$

where ρ_{proj} is the projected point cloud density in the image domain and comes from dividing the count of pixels with non-zero intensity values by the

total number of pixels in the chunk. $\kappa_{\max}$ denotes an upper bound to prevent the kernel from becoming too large for extremely sparse chunks. Subsequently, a global dilation with a kernel size of 2 is then applied to the whole image to improve its quality. Fig. 3.2 shows examples from before and after adaptive dilation.

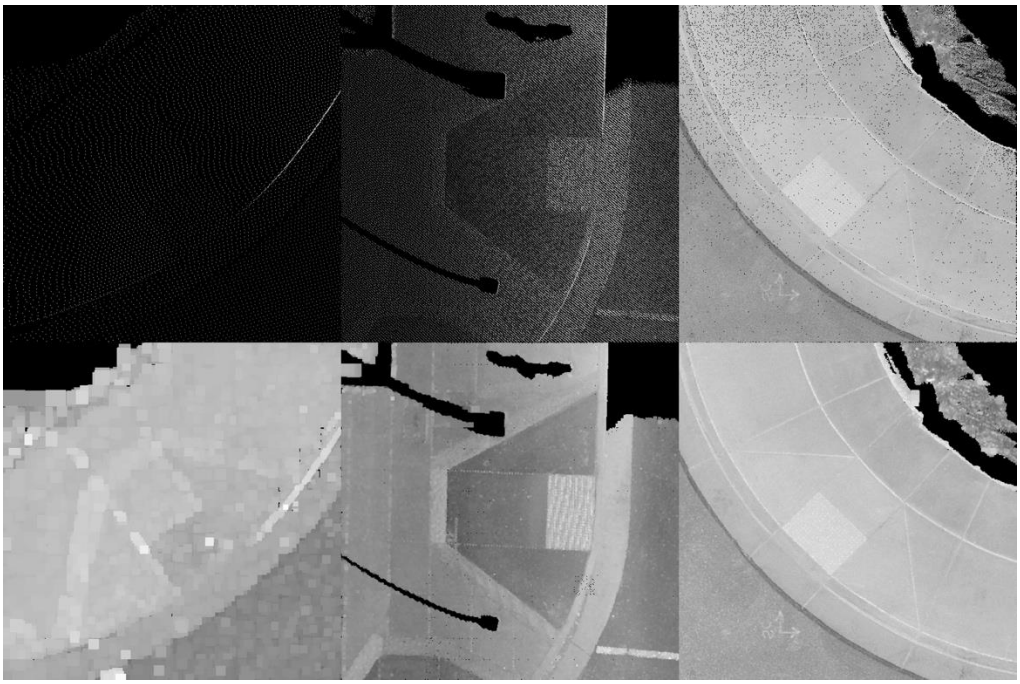

Figure 3.2: Grayscale images associated with the top-down view of point clouds of various densities before (top) and after (bottom) applying adaptive dilation. Adaptive dilation improves the 2D images in terms of contiguity for more accurate segmentation.

The ground-truth segmentation masks are similarly treated as grayscale images. Pixel intensities 0, 1, 2, etc. in a ground-truth mask image represent the associated class for each pixel. Once the point clouds are processed to create the images and their associated masks, they can be used to fine-tune a SAM for segmentation.

Segmentation with SAM

Our segmentation approach adapts SAM to enable data-efficient finetuning by leveraging its strong priors from pretraining on large image datasets. SAM consists of three main parts: an image encoder, a prompt encoder, and a lightweight mask decoder. The role of the prompt encoder is to receive any available region proposals and cues in the form of either a bounding box, prior mask, or text prompt. For our use case, the prompt is provided as a bounding box that contains the entire image. We modify the architecture so that the

same image and prompt encodings are provided to multiple independent mask decoder modules, each specializing in a specific segmentation class.

Ramp Decomposition into Constituting Components

The segmentation results from the previous step are converted back to 3D point clouds with class labels coming from the segmentation of the 2D projection (Figures 3.1 and 3.3a). The segments reflect a decent, though inadequate, annotation based on visuals, such as concrete joints. Ideally, however, the annotation should correspond to the geometric structure, i.e., the constituting primitive planes, rather than visual segments. In practice, there is often no clear geometric separation that perfectly matches the aesthetics, and in fact, the deviations are quite often significant (Figure 3.4). Additionally, as a result of performing 2D image segmentation, there could be points that are vertically separated but overlapping on the horizontal plane and thus mistakenly assigned to a particular segment. The segmentation quality paired with the fundamental distinction between aesthetic versus geometric components motivate further refinement of the segments before carrying out measurements.

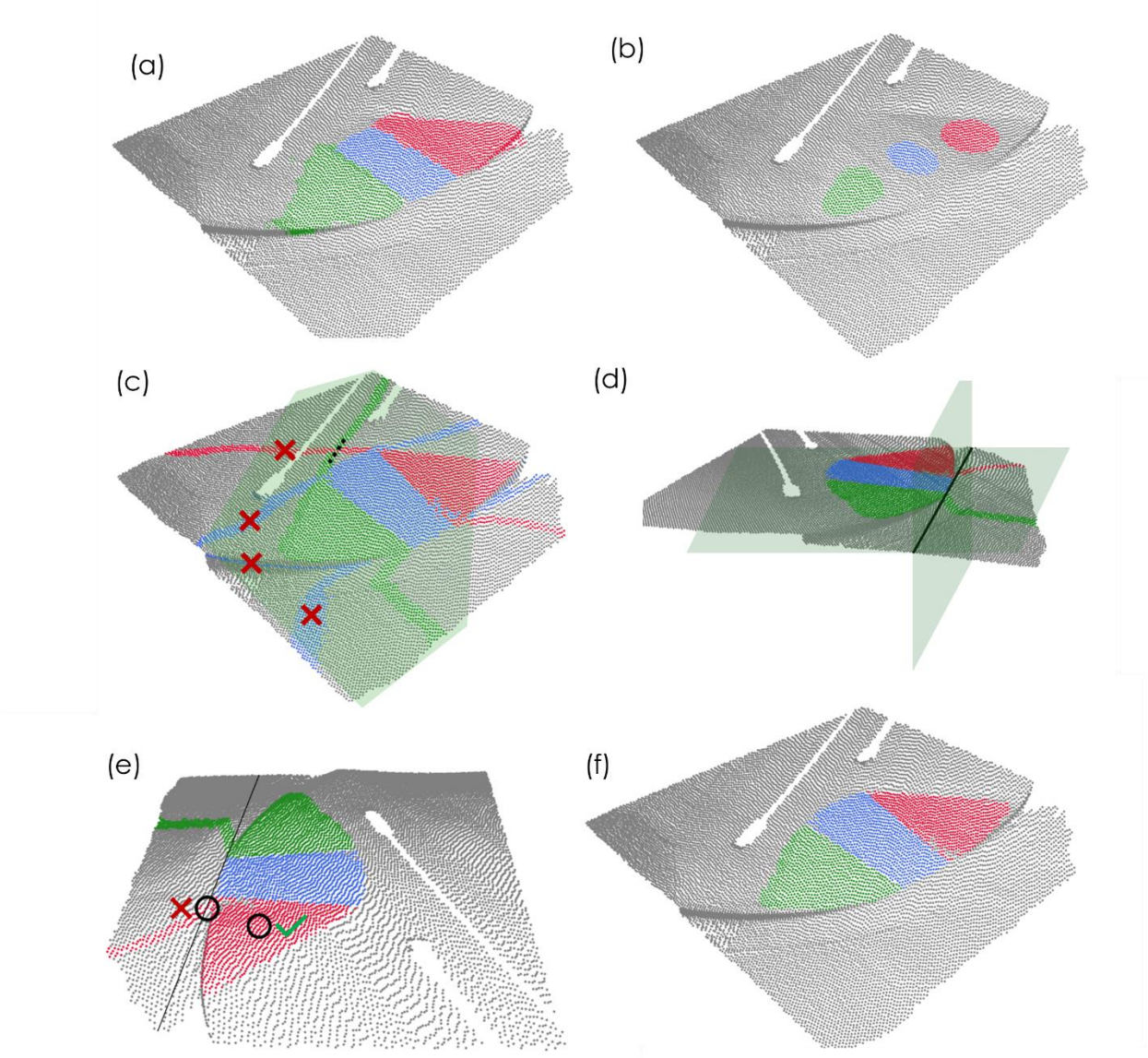

Figure 3.3: Geometric component decomposition. (a) Input component classifications, which are generally crude, agnostic to the vertical coordinate, and based solely on aesthetic features in the associated top-down 2D image. (b) Internal points for each component after applying One-Class Support Vector Machines (OCSVM). (c) Reclassification based on proximity to the internal coreset plane fits. The plane fit for the green component is shown. The planes are also used to cross-filter some of the ribbon-like artifacts. Red crosses denote ribbons associated with other components that can be cleaned out using the green plane. For the red ribbon, only the portion beyond the black dashed line is cleaned when filtering via the green plane. The process is repeated for all planes where incorrect labels are removed from the wrong half-space for each plane. (d) Using the bottom line fit to the center component, a vertical plane and its perpendicular plane passing through the line are drawn to filter out any component labels from the three incorrect spatial quadrants, leaving

component labels only in the correct quadrant (top left). (e) For each labeled point, its local plane fit is compared with the component plane fit. If the absolute cosine similarity between their normal vectors falls below a critical threshold, the point will be removed from the component. Two neighborhoods associated with a correct and an incorrect point are demonstrated respectively with a green check and a red cross next to them. (f) Final decomposition output after all steps, including iForest anomaly detection and final Density-Based Spatial Clustering of Applications with Noise (DBSCAN).

Figure 3.4: Photo of a side flare. The visual features often do not overlap with geometric attributes, and the deviations can be significant.

A key observation is that, while the segmentation result does not adequately serve as the component separation endpoint, it provides us with a valuable prior; specifically, we observe that the segmentation model almost always retrieves segments with correct internal points even for very poor-quality outputs. We base the geometric refinement on this premise, namely the inliers associated with a segment are assumed to belong overwhelmingly to the correct component. The second premise comes from the prototypical design of ADA ramps, which specifies that each surface that meets the standards is flat and should be well-approximated by a plane, with the possible exception of the gutter section for an ADA ramp with a curved roadside boundary.

We restrict the geometric refinement to the center ramp and warning surface as well as the side flares. Later in the text we will describe how the landing and gutter areas are handled. Further, considering that the warning

surface and the center ramp should be on the same plane, we combine these two components in our analysis when the warning surface is present. This means that we strive to identify and separate three components: ramp (plus warning surface if present), left flare, and right flare. Importantly, we seek a method that does not rely on manual intervention and customization for different ADA ramps.

Plane-Based Reassignment

We begin by removing the noisy points of each of the three components. To do this, we carry out an aggressive outlier detection and removal for each component. For this purpose, we employ one-class support vector machines (OCSVM) [27] with a radial basis function kernel $k(x, y) = \exp(-\gamma\|x - y\|^2)$ and a slack variable, ν. ν can be interpreted as an upper-bound on the fraction of outliers in each component.

Once a coreset is retrieved for each component (Figure 3.3b), it is used to fit a component plane associated with it. Following the plane fits, each initially excluded point is assigned to the component with the closest plane if its distance to that plane is less than a predefined threshold, *t*. It is noted that this reassignment is done for all points not belonging to a coreset. As a result, the ramp and flare boundaries now more accurately reflect the geometric surface characteristics. This step also removes the initially misclassified vertically separated points.

We note that, depending on the threshold, *t*, we might also remove points from each component that should nominally belong to them for measurement. However, we emphasize that the requirement for this stage is to have labels that can assist with defining references, such as corners and boundaries, and hence, useful in measurement, rather than correct class assignment for every point. During the measurement step, the potentially missed points will still contribute to the calculations.

Clean-Up

A consequence of reassigning points solely based on plane fits is that it often leads to substantial artifacts, external to the boundaries of ramp components, resulting from plane fits passing through the entire point cloud, including the sidewalk and street/gutter. These artifacts include continuous, typically ribbon-like extensions, above, below, and to the side of the ADA ramp area (Figure 3.3c). While anomaly detection and clustering algorithms like DBSCAN [28] can help mitigate this effect, we find it difficult, if not impossible, to pin down a set of universal DBSCAN hyperparameters that lead to robust performance across all or even a large fraction of ramps, making the choice impractical. This is due to the high variability across ramps and their surroundings. In the following discussion, we describe a primarily rule-based approach with minimal

hyperparameters that are much easier to set for robust performance across different ramps.

First, we highlight that at the component-level, an ADA center ramp and side flares constitute a (piecewise flat) concave surface by design. For components *i* and *j* of such an idealized surface, all points belonging to component *i* are only on one side of component *j* and vice versa (Figure 3.3c). Applying this observation to the assignments, we check whether each point assigned to component *i* is located in the correct half-space with respect to the plane fit for component $j \neq i$. A point is in the correct half-space if its coordinate along the normal vector of the plane fit associate with component *j* with respect to the initial coreset centroid, for *j* has the same sign as that of initial coreset *i* centroid along the same normal vector in the same coordinate system. We repeat this cross-check for all six possible combinations of components, and each time remove the label of points that reside on the wrong side of the plane.

This process usually removes mislabeled points and ribbons on the sidewalk and the curbs. However, prominent ribbons often misclassified as side flares remain on the street side as well as potential isolated points and small clusters farther from the ramp. The following steps further clean up the ramp area.

We fit a line to the bottom boundary of the center ramp component. The details of the bottom-line fitting will be described later. To this line, we first fit a vertical plane and then a generally tilted horizontal plane that is also perpendicular to the vertical plane. For each plane, we remove mislabeled points from the wrong half-space. As before, the correct half-space is determined by a sign match between the out-of-plane coordinate of a point and the ramp coreset centroid with respect to the plane's coordinate system (Figure 3.3d). This step removes most of the ribbons; some ribbon and miscellaneous points can still pass these filters and are nonetheless problematic for reference point detection.

Next, we note that incorrect ribbon points locally reside on surfaces with local plane fits different from their assigned component plane fit. We can leverage this fact to further trim them away. To trim each component, we first obtain the nearest neighbors of each point that are within a ball of radius *r* centered at the point (Figure 3.3e). The value *r* should be selected to accommodate varying point cloud densities and to ensure that the ribbon points do not dominate the neighborhood. For each point, we can, therefore, obtain a plane fit to its local neighborhood. If the absolute value of cosine similarity between a local normal vector and component normal vector fall below a defined similarity threshold s_{crit}, the point can be dismissed as a misclassification because its local plane does not adequately resemble the component plane.

A caveat here is that, for the correctly classified points at the boundaries of ramp or flares, a local neighborhood will contain a significant number of incorrect points from the adjacent component and/or the non-ramp area. To

mitigate this error, we can bias the local neighborhood such that only neighbors to the correct side of a point are considered. Empirically, we define the correct side of the point as one which is closer or equal in distance from the initial component coreset centroid compared to the point. Biasing the neighborhood in this way leads to a neighborhood of completely or overwhelmingly correct points for the component boundaries, leading to a local fit that closely resembles the component plane fit.

Following this stage, we are typically left with a sparse amount of outliers. To remove them, we employ Isolation Forest (iForest) [29]. While the optimal value for the contamination parameter (i.e., the expected proportion of outliers) depends on the specific ramp, we find that the search zone for this parameter is relatively small and that a fixed small value can yield a good performance across different ramps. Even if manual intervention for parameters adjustment is desired, it is now an interpretable and decoupled parameter that can be set more easily.

Finally, we apply DBSCAN to the set of all labeled points in case, after the anomaly detection, there still remain some isolated incorrect points. We remove the label of any points not belonging to the largest cluster, which corresponds to the combined ramp and flares. An example output for this process is shown in Figure 3.3f.

Reference Point Detection

Reference points are used to extract the measurements. As such, accurate detection of reference points is crucial for geometric measurements. Despite our extensive pre-processing and outlier detection efforts in previous steps, the point clouds may still contain noise and outliers, which can be affected by their varying density. To address these challenges, we present a robust reference point detector that combines a novel Score-Based Line Fitting (SBLF) algorithm with geometric constraints to identify the key points that fully define the ramp structure. Figure 3.5 offers a thorough example demonstrating the multi-stage reference point detection pipeline.

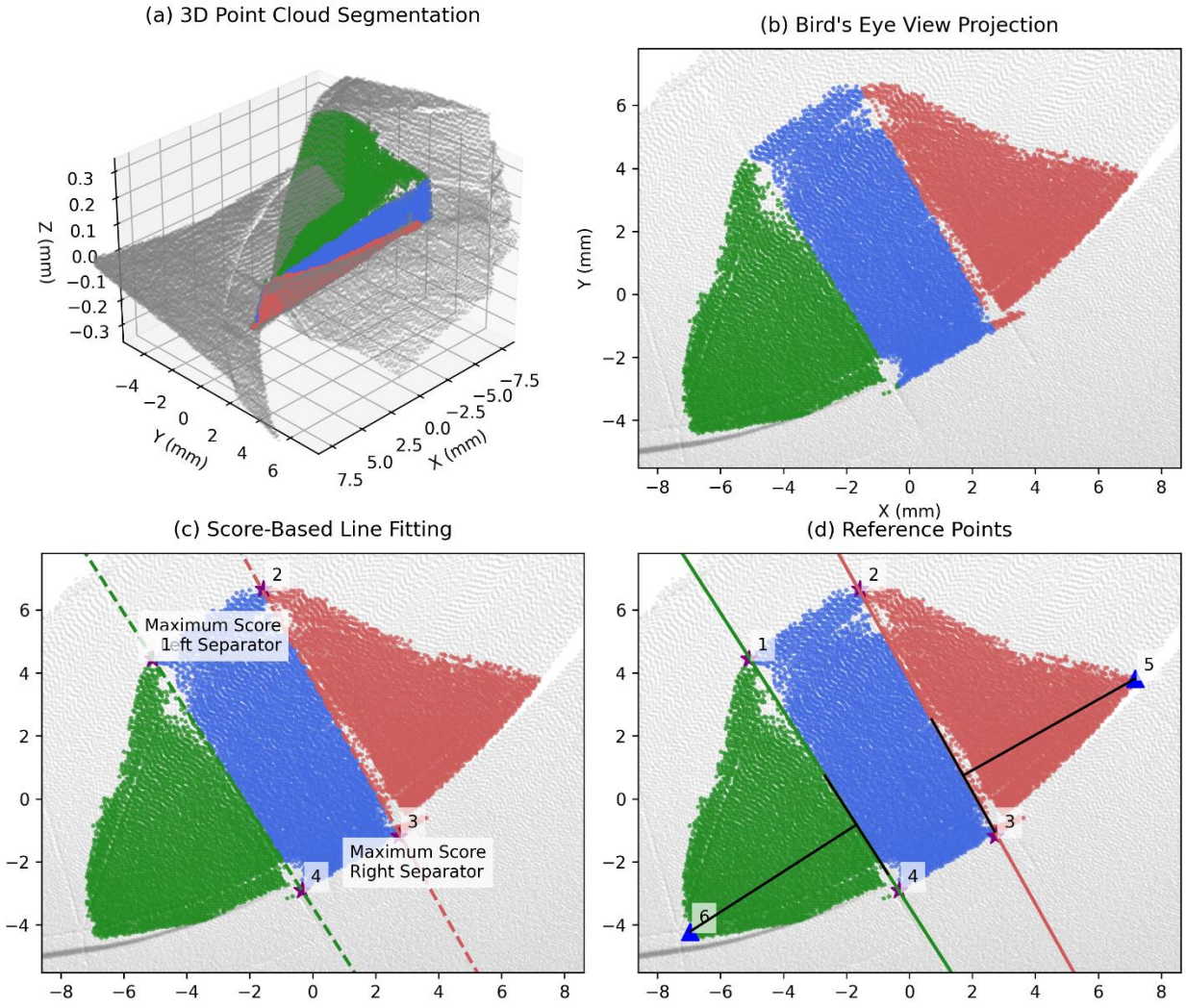

Figure 3.5: Proposed multi-stage reference point detection pipeline. First, we convert the 3D point clouds to bird's eye views (b) to enable efficient 2D processing while preserving geometric relationships. We then employ our novel SBLF algorithm (c) to identify component boundaries by maximizing separation scores between point classes, eliminating the need for parameter tuning while maintaining robustness to outliers. Finally, we leverage structural priors to detect additional reference points by identifying locations with maximum projection distance to the fitted boundaries (d), ensuring comprehensive capture of the ramp geometry.

Score-Based Line Fitting

We first convert the latest filtered and decomposed 3D point cloud data to a top-down 2D view. The SBLF algorithm is designed specifically for identifying clear geometric boundaries between point cloud segments while maintaining robustness to outliers and sparsity. Unlike traditional approaches, such as

RANSAC [30] or SVM [31], that can be sensitive to noise and the selection of a hyperparameter, SBLF operates by exhaustively evaluating potential boundary lines formed by point pairs, focusing on optimal separation between component regions without requiring parameter tuning.

Given two point-sets *A* and *B* to be separated, and a candidate point set *C* from which to sample line-forming points, SBLF computes a separator line by evaluating all possible lines formed by pairs of points from the convex hull of *C*. For each candidate line *l*: $ax + by + c = 0$, we calculate a separation score:

$$
\begin{aligned}
S(l) &= S_A + S_B \\
S_A &= |\{p \in A : ap_x + bp_y + c < 0\}| \\
S_B &= |\{p \in B : ap_x + bp_y + c > 0\}|
\end{aligned}
$$

where $|\cdot|$ denotes set cardinality. The algorithm returns the line coefficients (*a*, *b*, *c*) that maximize this score, along with the optimal orientation. The separation score is computed by counting the total number of correctly classified where 'correct' means points from group A lie on one side while points from group B lie on the other. This approach naturally favors lines that create a clean geometric division between components while being robust to noise without need for parameter tuning.

Component Separation

The reference point detection process begins by identifying the primary geometric boundaries of the ramp using SBLF to compute the separator line between different ramp components. For example, to fit the bottom line, we apply SBLF to separate the ramp points and non-ramp points using candidate points from the ramp. For boundaries between flares and the main ramp, to enhance accuracy and robustness against non-uniform point distributions near component boundaries, we compute two candidate separator lines: one using only the flare region points for candidate points, and another using only the main ramp points. The final separator line is then determined by averaging the coefficients of these two lines using the following equation:

$$
\begin{aligned}
l_f &= \mathrm{SBLF}(P_{flare}, P_{middle}, P_{flare}) \\
l_m &= \mathrm{SBLF}(P_{flare}, P_{middle}, P_{middle}) \\
l &= \frac{l_f + l_m}{2}
\end{aligned}
$$

Corner Point Detection

Using the separator lines, we identify six reference points that characterize the ramp geometry. Four points are computed as intersections between the separator lines and the ramp's convex hull. The remaining two points mark the peaks of the left and right flares, identified by maximum projection distance:

$$p_{flare} = argmax_{p \in P_{flare}} \frac{|ap_x + bp_y + c|}{\sqrt{a^2 + b^2}}$$

Finally, we map the 2D reference points back to 3D space through nearest neighbor search in the original point cloud, preserving elevation information critical for subsequent geometric measurements.

Quality Assurance

To improve the reliability of our measurements and prevent propagation of errors in the pipeline, we first remove ramps with insufficient point cloud density. Specifically, any ramp with an average point density below a predefined threshold is excluded to avoid inaccuracies in slope or length estimation due to sparsely sampled or incomplete surface geometry. We apply density-based filtering at this stage because, for a more relevant and accurate density estimation, we need to restrict the estimate to the individual ramp asset and not an entire point cloud. Further, density in terms of points per area is a better and more robust estimate compared to volumetric density, and as such, density-based filtering is applied once we geometrically refine the components.

The surface density estimate is obtained by first projecting the points for each component on the corresponding plane-fit and then obtaining the area of the convex hull of the component projection. Density can then be estimated by dividing the total number of considered points by the sum of the areas. While we can perform this analysis for the final components, we leverage the initial OCSVM-identified coresets for the 2D density estimate instead in order to avoid potential error propagation, considering the coresets still incorporate many points and a substantial surface area sufficient for obtaining a robust estimate.

Additionally, to ensure geometric consistency in corner detection, we apply two filtering methods. First, we analyze the distribution of angles at the identified corners and exclude ramps where an angle deviated more than three standard deviations from the dataset mean. This analysis works as a statistical outlier detection step where ramps with atypical corners are removed. Second, we enforce a parallelism constraint on the top and bottom edges of the center ramp. This constraint is informed by our a priori knowledge about the construction of ADA ramps and can be adapted to other assets. This constraint

is critical because these edges are later used to approximate the landing and gutter regions as well as their corresponding measurements.

Measurement Extraction

Measurement extraction mainly relies on the extracted reference points, i.e., corners (previous steps). For ADA ramps quantities, such as slope and width, are needed. The exact measurement procedure should be aligned with the approved state of practice. As such, although in some cases one may think of better or more efficient ways of measuring a certain value, it is important to stick to approved guidelines to ensure that extracted measurements are legally reliable in the case of future litigation. Any changes in the measurement procedure should be made with the approval of legal entities, e.g., in this case within Caltrans. As such, meaningful geometrical references should follow the measurement extraction guidelines described in Caltrans official documents such as that listed in Appendix A and indicated in Figure 3.6.

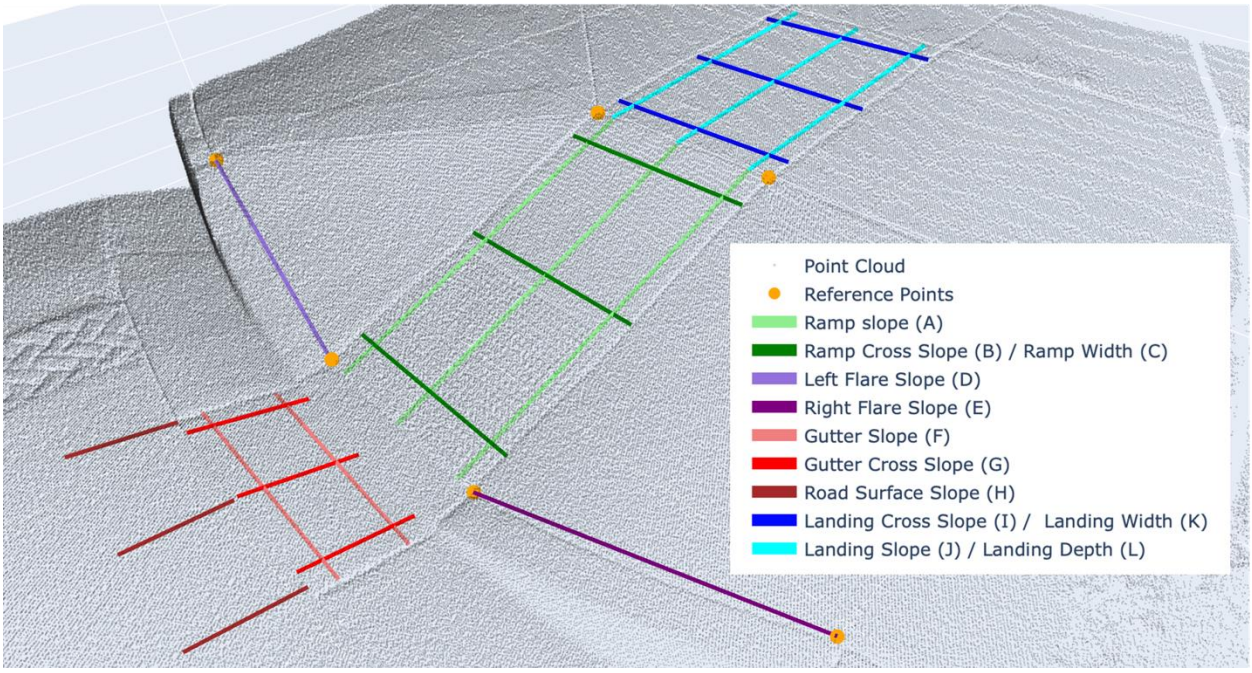

Figure 3.6: Ramp and gutter layout with corner points and measurement lines. Corner points define points necessary for measurements, and measurement lines indicate the guidelines on where/how slopes and widths are expected to be measured for the ramp.

Center Ramp Measurement

As seen in Figure 3.6, the center ramp consists of four corner points and the objective is to measure ramp slope (A), ramp cross slope (B), and ramp width (C).

Generate Reference Lines: Following input from the project panel, we use the following procedure to extract measurements. Each edge of the center ramp is divided into 10 equal segments. To create initial reference lines, points at *1/10*, *5/10*, and *9/10* of each point is connected with their corresponding points on the opposite side.

Refine Reference Lines: For each reference line, we collect the 300 nearest points (to the line) from the point cloud. Using these points, we fit a new line via Principal Component Analysis (PCA) to ensure that the point cloud geometry is accurately reflected. Once the line is fitted, points that are more than one-quarter inch away from the line are filtered out, adhering to Caltrans' measurement guidelines. This process of line fitting and point discarding is repeated iteratively until no points are removed. The final fitted line is used for slope and width measurements, ensuring that it conforms to the ramp geometry and the established guidelines.

Slope and Width Measurement: Given the final measurement lines, we extract the slope and width of the center ramp. For each fitted line, two endpoints are determined using the boundaries of the center ramp. Then the slope is calculated as the ratio of vertical to horizontal displacement between the endpoints of the line. To calculate the width, we measure the distance between the two endpoints of each fitted line. These steps collectively ensure that the slope and width are measured based on the actual surface geometries of the ramp represented in the point cloud rather than simply relying on two points on the ramp boundaries for calculating slope and width.

Flare Measurement

The left and right flares each consist of three corner points (Figure 3.6). The objective is to measure the slopes of D and E, which define the running slopes of the two flares. We generate a reference line from the two bottom corner points. Using these points, we apply the same iterative process outlined for the center ramp measurement: collecting nearby points from the point cloud, fitting a line using Principal Component Analysis (PCA), and filtering points more than one-quarter inch away from the line. The final fitted line is then used to calculate the running slope of the two flares.

Landing and Gutter Measurement

In our analysis pipeline, the landing and the gutter areas are approximated by imposing their minimal standard geometric definition directly onto the point cloud. For the landing area, we estimate the top landing boundary by extending the side edge of the center ramp outward and searching perpendicularly from the top edge. This search is performed iteratively, extending the edge by a small fixed length at each step. At each iteration, the endpoint of the extended line is projected vertically (in the z-direction) onto the point cloud, forming a sloped line between the projected point and the top corner of the center ramp. As the search progresses, the slopes of these consecutive lines are continuously monitored. When the difference between two consecutive slope calculations is larger than a pre-defined threshold, the algorithm marks the last valid projected point before the discontinuity as the top corner point of the landing. This procedure is executed on both sides of the ramp to determine the full top landing boundary. Similarly, the gutter area is approximated using the bottom edge of the center ramp as reference. Using the boundary points derived from the approximation, we can calculate the slopes and lengths following the same measurement steps used for other ramp components.

Chapter 4:
Experiments

Dataset and Implementation Details

Detection

The annotated ramp detection dataset consists of 1604 images with 3217 bounding boxes. Each image represents a crop of a street-level point cloud, projected top-down and converted into grayscale using normalized LiDAR intensity values. Example images are shown in Figure 4.1.

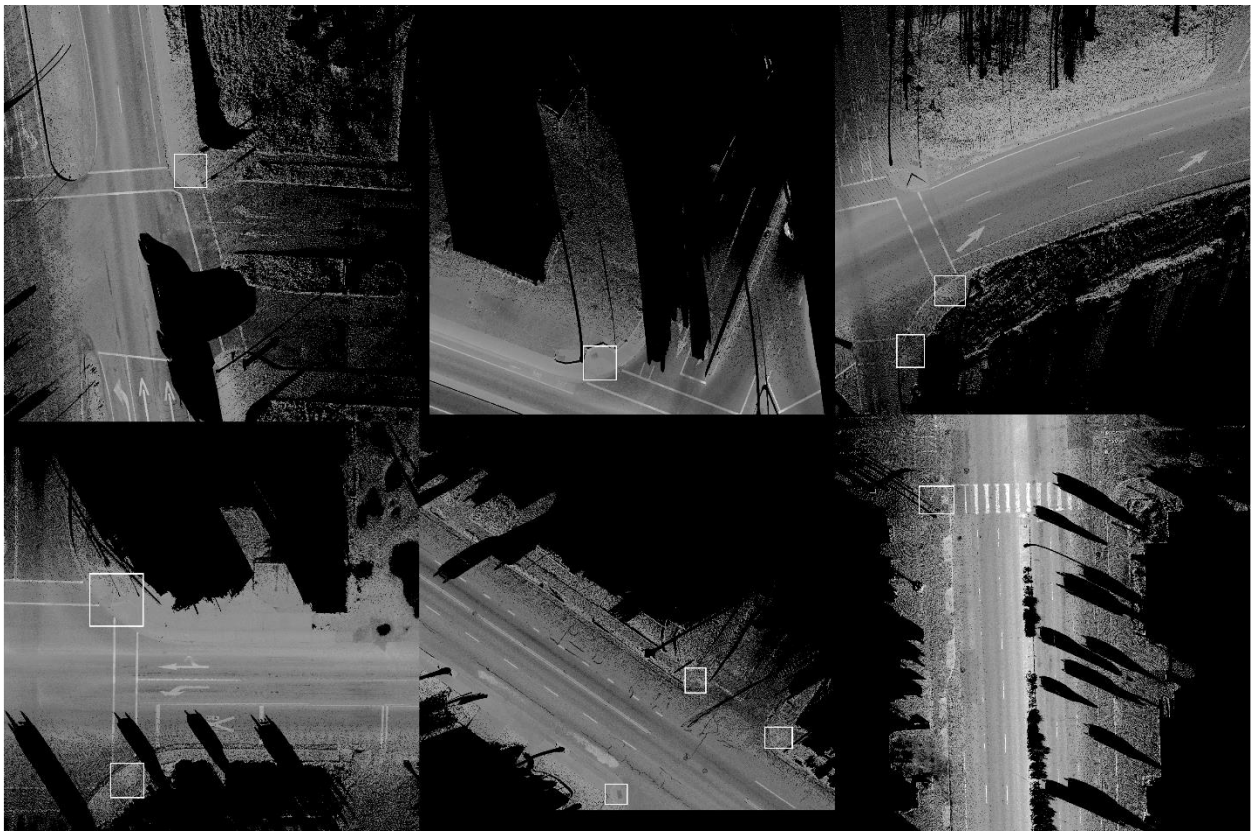

Figure 4.1: Example images in ramp detection dataset. Each image represents a crop of a street-level point cloud and may contain ramps.

Our DeTR models were trained on an 8:2 split dataset with a batch size of 32 for 500 epochs. The Adam optimizer was used with a learning rate of 10^{-5} and a weight decay of 10^{-4}.

Segmentation

The available data, including street/block-level point clouds, were manually annotated by a third-party firm which delineated various segments of the ramp. The manual annotation output assigns to each unlabeled point one of six classes: center ramp, warning surface, left/right flares, landing, and gutter. The individual asset point clouds are then extracted and converted to 512 by 512 grayscale images. To apply adaptive dilation, we divide each image into 32 by 32 chunks and consider $\kappa_{max} = 50$.

The input images and masks for SAM are resized to 256 by 256. Masks are represented as one-hot tensors with a channel dimension where different channels represent different classes. The mask represents the ground truth pixel classification for each segment of the ramp. The prediction for each channel is generated by its associated mask decoder. The model is initialized with pretrained weights. The image and prompt encoder weights are frozen during training and only the decoder modules are finetuned. The training process employs a hybrid loss consisting of Dice loss and cross-entropy loss with equal weights, with sigmoid activation applied to the decoder outputs. The model with a ViT Base [32] image encoder is trained on 1,541 samples with a batch size of 16 for 100 epochs. Adam optimizer is used with a learning rate of 10^{-5} and a weight decay of 10^{-4}.

Geometric Component Decomposition

After retrieving point cloud class assignments, we set ν = 0.7 for extracting internal component points for each of the three components (left/right flares, combined center ramp/warning surface). We find this to be a conservative value that consistently yields accurate candidate points. We choose the kernel coefficient γ based on the default heuristic where $\gamma = 1/3{\sigma_x}^2$ in which ${\sigma_x}^2$ is the variance of the (centered) input point cloud coordinates and 3 in the denominator represents the space dimensionality.

We set a threshold of t = 0.6 in. We also set the neighborhood ball radius to r= 3 in, and the critical absolute cosine similarity to $s_{crit} = 0.999$. We perform iForest with 100 trees and a contamination of 0.02. Finally, we set the DBSCAN distance parameter $\varepsilon = 2$ ft.

Detection Performance

The ramp detection results are organized in Table 4.1. The two DeTR models demonstrated superior performance compared to the Faster R-CNN model. Specifically, the DeTR model with the ResNet-50 backbone achieved a mAP@50 of 0.866 and a recall@50 of 0.765, while the ResNet-101 backbone further improved these metrics to a mAP@50 of 0.873 and a recall at IoU ≥ 50 of 0.771.

Table 4.1 Detection Performance on Test Dataset

Model	mAP@50	mAP@50:95	Recall at IoU $\geq$ 50
Faster R-CNN (ResNet-50)	0.538	0.428	-
DeTR (ResNet-50)	0.866	0.661	0.765
DeTR (ResNet-101)	**0.873**	**0.684**	**0.771**

The use of overlapping crops during inference enhances overall detection reliability because the same ramp is seen multiple times under varying contexts or backgrounds. This strategy ensures that when a ramp is not detected in one crop, it can still be identified in other crops, reducing the chances of missed ramps.

Survey Accuracy

The trained SAM model was tested on 61 visualized point clouds. The average Dice score across the six asset regions (excluding the background class) was calculated as 0.857, with the highest score attributed to flare segment (0.890) and the lowest score corresponding to the gutter (0.822). Detailed results are provided in Table 4.2.

Table 4.2 Segmentation Performance across Asset Regions

Segment	Dice Score	Segment	Dice Score
Center Ramp	0.842	Warning Surface	0.882
Right Flare	0.880	Left Flare	0.890
Landing	0.822	Gutter	0.822
Mean (Excluding Background)			0.857

By analyzing angles derived from corner points, we identified anomalies that deviated more than three standard deviations from the mean. This method resulted in the exclusion of four ramps from the dataset, which were identified as outliers due to significant deviations in their geometric properties.

Figure 4.2 illustrates an example of excluded outliers, which shows irregular segmentation or corner placements that do not conform to expected patterns. These ramps exhibit irregular segmentation or corner placements that deviate significantly from expected geometric patterns. As such, the proposed methodology will flag the ramps for manual investigation by experts.

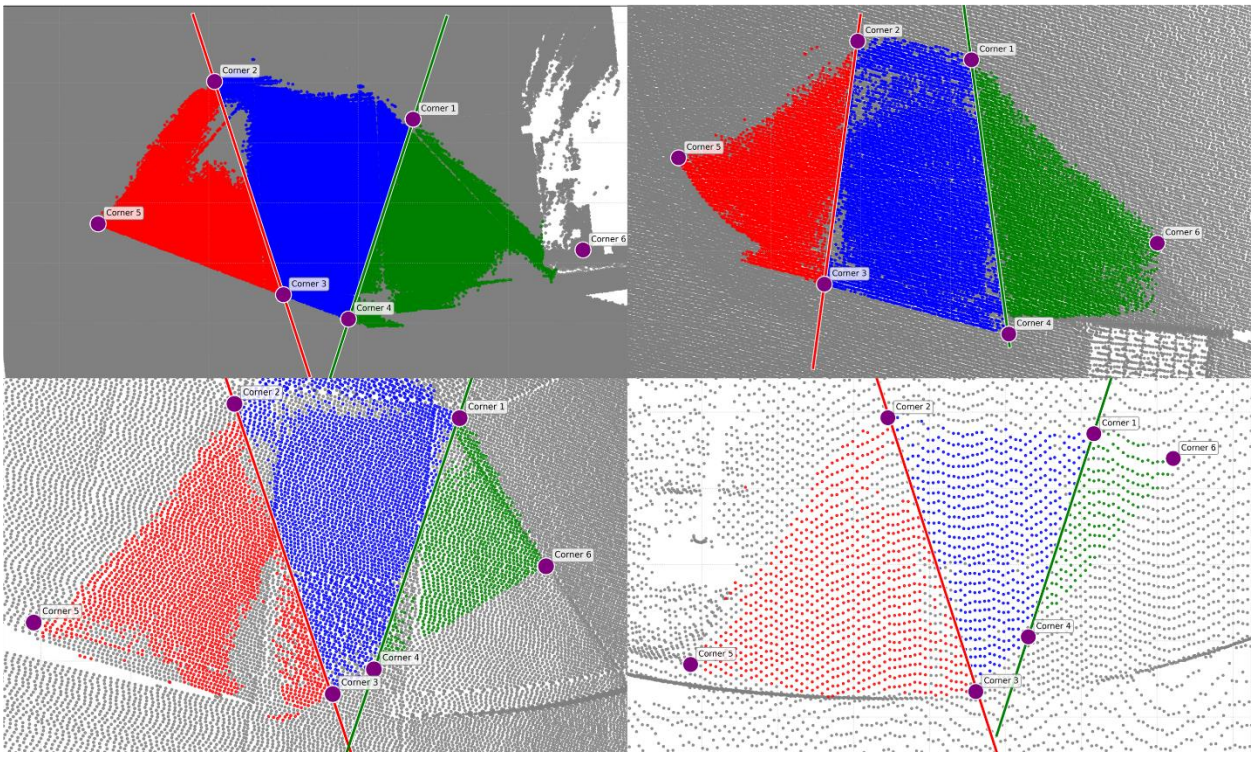

Figure 4.2: The excluded outlier ramps.

For quality control, density-based filtering is applied to the 46 ramps with components geometrically refined in the previous step. Eleven ramps were disqualified due to insufficient point cloud resolution, leaving 35 ramps for further analysis. Among these, another four were disqualified per angle-based statistical filter using a three-standard-deviation threshold, and 11 were removed by the parallelism constraint on opposing ramp edges. In total, 20 ramps passed both stages of quality control and were retained for the final evaluation.

To evaluate our automated measurement pipeline, we compared the ADA compliance assessment results against manual measurements (see Appendix A for manual field measurement details). We conducted field measurements on 16 ramps in the City of Woodland, which were selected from the subset that passed quality control and were accessible based on proximity to UC Davis campus and ease of access to our team.

The comparison was performed across 12 primary ADA ramp features, each of which is used by Caltrans in compliance assessment as illustrated in Table 4.3. Each feature comprises one or more measurements (e.g., A1–A3, B1–B3, etc.), resulting in a total of 31 sub-feature measurements used for detailed evaluation. For each feature, we calculated the average measurement difference and compliance consistency between the automated and manual assessments.

Table 4.3 Measurement Labels and Compliance Standards

Feature	Compliance
A - Ramp Slope (%)	$\leq 7.7\%$
B - Ramp Cross Slope (%)	$\leq 1.7\%$
C - Ramp Width (inches)	$\geq 49.75"$
D - Left Flare Slope (%)	$\leq 9.2\%$
E - Right Flare Slope (%)	$\leq 9.2\%$
F - Gutter Slope (%)	$\leq 1.7\%$
G - Gutter Cross Slope (%)	$\leq 5.2\%$
H - Road Surface Cross Slope (%)	$\leq 5.2\%$
I - Top Landing Cross Slope (%)	$\leq 1.7\%$
J - Top Landing Slope (%)	$\leq 1.7\%$
K - Top Landing Width (inches)	$\geq 49.75"$
L - Top Landing Depth (inches)	$\geq 49.75"$

Figure 4.3 illustrates the absolute mean differences between our automated measurements and manual measurements for each ADA ramp feature. For slope-based features (left), the differences remain relatively small, generally under 1%, indicating strong alignment between automated and manual processes. The largest slope deviation was observed for the road surface slope (feature H), at approximately +0.54%. For length-based features (right), greater discrepancies were found, particularly in landing depth (feature L), where manual annotations tended to exceed automated estimates by an average of 7.88 inches. This discrepancy is likely due to the approximation involved in estimating the top landing line wherein small artifacts or irregularities in the surface geometry may lead to inconsistent depth calculations between manual and automated methods.

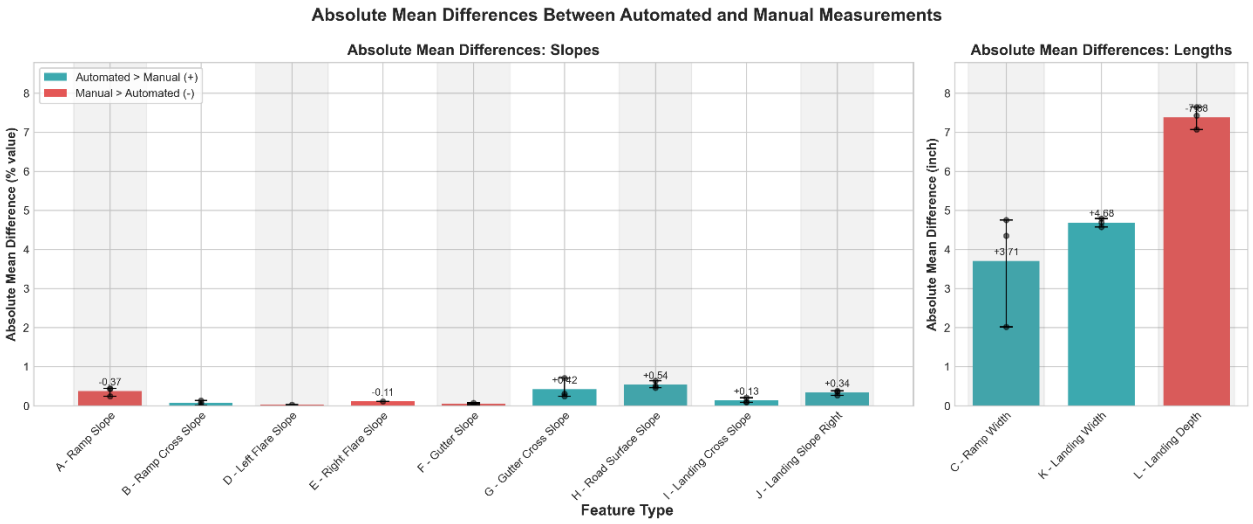

Figure 4.3: Absolute mean differences between automated and manual measurements across ADA ramp features. Differences are separated into slope-based (% value) and length-based (inches) measurements. Positive values (blue) indicate overestimation by the automated pipeline, while negative values (red) indicate underestimation.

Since the final decision on ADA compliance is binary, we further evaluated the practical utility of our system by comparing binary compliance assessments from the automated pipeline with those obtained through manual evaluation. Figure 4.4 highlights the number of compliance conflicts per feature type. The majority of disagreements occurred in width-based features, particularly landing width (six cases) and ramp width (four cases) followed by ramp slope (three cases). These features are often sensitive to boundary definitions, and the discrepancies can arise from local surface irregularities not captured uniformly in manual inspections, especially when concrete lines or transitions are ambiguous. Such inconsistencies underscore the benefit of a consistent, geometry-based approach for surveying as used in our automated pipeline. It should be noted that in cases where automated assessment flags a ramp as non-compliant, manual follow-up can be limited to the specific feature(s) that failed quality checks or compliance thresholds. This targeted verification further reduces the overall manual effort compared to traditional full-ramp assessments.

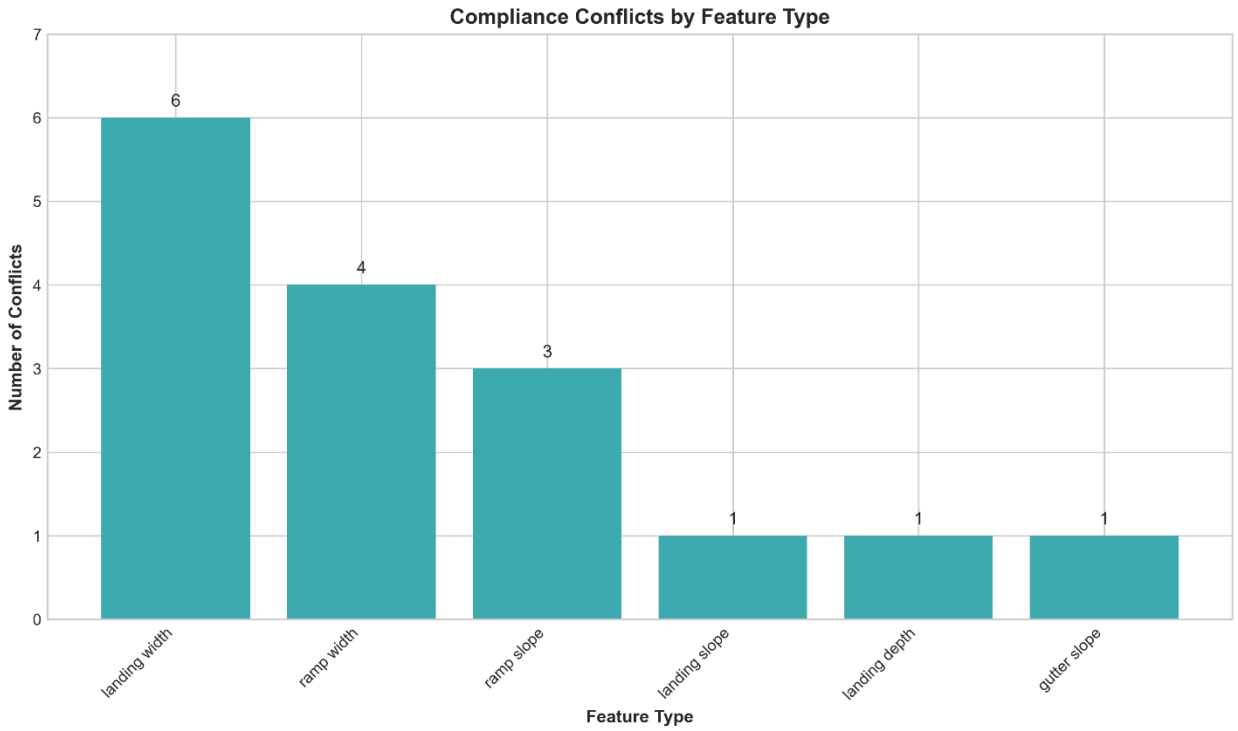

Figure 4.4: Analysis of compliance consistency between automated and manual ADA ramp assessments. The bar chart shows the number of compliance conflicts per feature type with the most frequent discrepancies observed in landing width and ramp width.

To account for human error in manual measurements or inaccuracies due to inconsistent point clouds, we define a tolerance margin around the compliance threshold to assess whether the disagreement is significant. Let the margin be defined as:

$$\delta = T \times \frac{p}{100}$$

Where p is our defined margin percentage (e.g., $p = 5$). We define the acceptable range as:

$$[T - \delta,\ T + \delta]$$

If both the automated measurement A and the manual measurement M fall within this range:

$$T - \delta \leq A \leq T + \delta \quad \text{and} \quad T - \delta \leq M \leq T + \delta$$

then the disagreement is considered within tolerance and not treated as a significant conflict.

We set tolerance margins of 5% and 10% to evaluate the robustness of compliance consistency. The left chart in Figure 4.5 shows that 87.9% of feature

assessments were consistent with manual measurements. Among the remaining 12.1%, the breakdown (right) shows that most discrepancies (13 out of 16) fell within a 5% tolerance margin, and only a single instance exceeded a 10% margin. This analysis suggests that a majority of the conflicts stem from minor, and likely acceptable, human variation, rather than fundamental measurement errors in the automated system. The results demonstrate that our method remains robust and reliable even under modest margin thresholds.

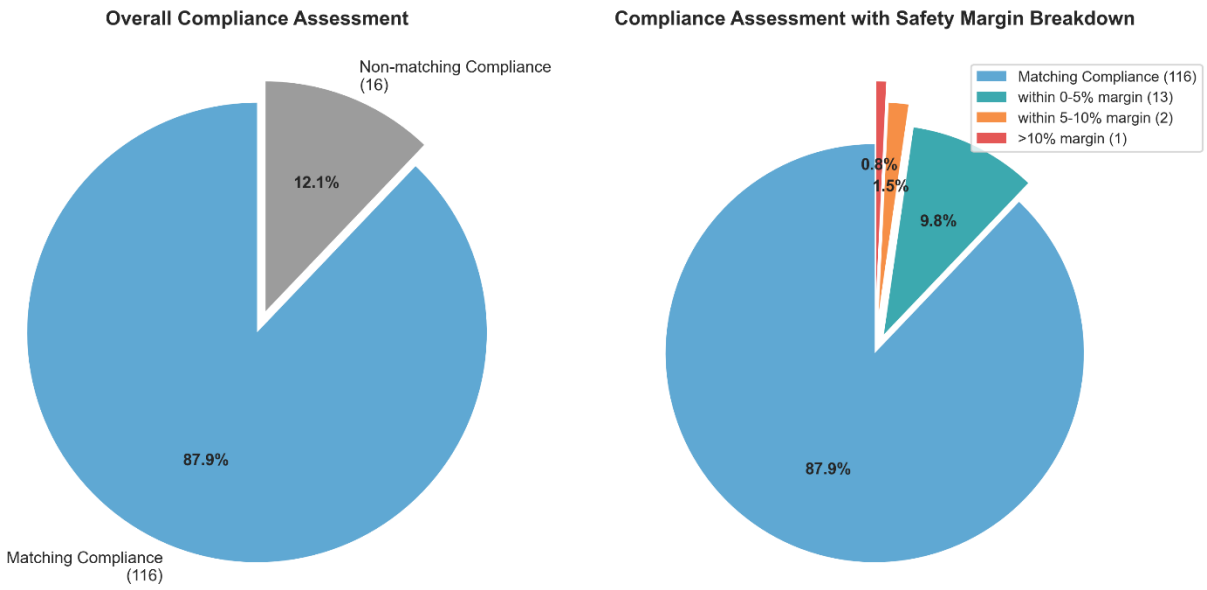

Figure 4.5: (Left) Overall compliance agreement between automated and manual assessments without any tolerance margin applied. (Right) Breakdown of the 12.1% non-matching compliance results after introducing progressive tolerance margins of 5%, 10%, and beyond. Most conflicts fell within a 5% margin, suggesting that minor deviations in manual measurement could explain a majority of the disagreements.

Chapter 5:
Discussions and Future Work

While our automated pipeline demonstrates strong results, there are several limitations worth considering. First, the accuracy of the automated measurements is inherently dependent on the quality and density of the input point clouds. Sparse or noisy data, often caused by occlusions, or poor sensor calibration, can lead to misidentification of ramp boundaries or make slope/length measurements error prone. Although we apply outlier filtering, this step cannot fully mitigate the effects of incomplete data. Second, several ramp components, such as landing edges and the gutter transitions, may lack explicit visual cues in the field. In those cases, both manual and automated methods rely on approximations. In particular, our method uses geometric projections and slope-based heuristics to infer feature boundaries, which may diverge from human interpretation in ambiguous regions where markings are faded or surface continuity is disrupted. One last limitation is that while our approximated landing and gutter boundaries work reasonably well, they rely on geometric regularity and slope transitions that may not generalize well to highly irregular ramp designs. In such cases, the automated system might either fail to detect the intended boundary or assign it incorrectly.

We further note that field measurements are not entirely geometric, and in practice, ramp visual cues, such as boundaries, factor into the measurements. For instance, ramp cross slope direction is defined relative to the visible ramp boundaries. Consequently, there exists an inherent difference between the field measurements and those obtained by the proposed method when separation of ramp components per visual references and local surface topography do not align. As discussed earlier, we used visual segmentation to retrieve a relatively crude prior for geometry-based component decomposition. Given the aforementioned inherent difference, if a closer match with field measurements is desired, it can be obtained by relying more closely on the visual segmentation output rather than geometric modeling. Of course, as previously stated, an output based on visual segmentation would not align with actual geometric features as closely, which is the de facto motivation for the measurements.

Future research directions include extending the pipeline to handle partially occluded assets and further exploring multimodal data, for instance combining LiDAR with high-resolution images.

References

[1] J. K. Liu, R. Qin, and S. Song, 'Automated Deep Learning-Based Point Cloud Classification on USGS 3DEP Lidar Data Using A Transformer', in *IGARSS 2024 - 2024 IEEE International Geoscience and Remote Sensing Symposium*, 2024, pp. 8518–8521.

[2] H. Kim and C. Kim, 'Deep-Learning-Based Classification of Point Clouds for Bridge Inspection', *Remote Sensing*, vol. 12, no. 22, 2020.

[3] B. L. Bhavesh Kumar Gaurav Pandey and S. C. Misra, 'A framework for automatic classification of mobile LiDAR data using multiple regions and 3D CNN architecture', *International Journal of Remote Sensing*, vol. 41, no. 14, pp. 5588–5608, 2020.

[4] R. Varghese and S. M., "YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness," 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), Chennai, India, 2024, pp. 1-6, doi: 10.1109/ADICS58448.2024.10533619.

[5] S. Ren, K. He, R. Girshick, and J. Sun, 'Faster R-CNN: Towards real-time object detection with region proposal networks', *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[6] M. Liu and J. Niu, 'BEV-Net: A bird's eye view object detection network for LiDAR point cloud', in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 5973–5980.

[7] Z. Li, S. Lan, J. M. Alvarez, and Z. Wu, 'BEVNeXt: Reviving Dense BEV Frameworks for 3D Object Detection', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20113–20123.

[8] L. Peng, Z. Chen, Z. Fu, P. Liang, and E. Cheng, 'Bevsegformer: Bird's eye view semantic segmentation from arbitrary camera rigs', in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 5935–5943.

[9] S. Shi, X. Wang, and H. Li, 'Pointrcnn: 3d object proposal generation and detection from point cloud', in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 770–779.

[10] C. R. Qi, O. Litany, K. He, and L. J. Guibas, 'Deep hough voting for 3d object detection in point clouds', in *proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9277–9286.

[11] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, 'Deep Learning for 3D Point Clouds: A Survey', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4338–4364, 2021.

[12] H. Peng *et al.*, 'Semantic Segmentation of Litchi Branches Using DeepLabV3+ Model', *IEEE Access*, vol. 8, pp. 164546–164555, 2020.
[13] C. Wang, P. Du, H. Wu, J. Li, C. Zhao, and H. Zhu, 'A cucumber leaf disease severity classification method based on the fusion of DeepLabV3+ and U-Net', *Computers and Electronics in Agriculture*, vol. 189, p. 106373, 2021.
[14] O. Ronneberger, P. Fischer, and T. Brox, 'U-Net: Convolutional Networks for Biomedical Image Segmentation', in *Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015*, 2015, pp. 234–241.
[15] A. Kirillov *et al.*, 'Segment Anything', in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 4015–4026.
[16] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, 'PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space', *arXiv [cs.CV]*. 2017.
[17] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, 'KPConv: Flexible and Deformable Convolution for Point Clouds', *arXiv [cs.CV]*. 2019.
[18] Q. Hu *et al.*, 'RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds', *arXiv [cs.CV]*. 2020.
[19] Y. Lyu, X. Huang, and Z. Zhang, 'Learning to Segment 3D Point Clouds in 2D Image Space', *arXiv [cs.CV]*. 2020.
[20] C.-K. Yang, M.-H. Chen, Y.-Y. Chuang, and Y.-Y. Lin, '2D-3D Interlaced Transformer for Point Cloud Segmentation with Scene-Level Supervision', *arXiv [cs.CV]*. 2024.
[21] H. Guo *et al.*, 'SAM-guided Graph Cut for 3D Instance Segmentation', *arXiv [cs.CV]*. 2024.
[22] B. D. Planet, 'Caltrans Settles ADA Lawsuit with $1.1 Billion Commitment', 2009. [Online]. Available: https://www.berkeleydailyplanet.com/issue/2009-12-23/article/34346. [Accessed: 09-Jan-2025].
[23] LAist, 'L.A. to Pay $1.4 Billion in Sidewalk Repairs, Settling Largest Disability Access Lawsuit', 2015. [Online]. Available: https://laist.com/news/kpcc-archive/l-a-to-pay-1-4-billion-in-sidewalk-repairs-settlin. [Accessed: 09-Jan-2025].
[24] L. A. at Work, 'Willits v. City of Los Angeles Sidewalk Settlement Announced', 2015. [Online]. Available: https://legalaidatwork.org/willits-v-city-of-los-angeles-sidewalk-settlement-announced-2/. [Accessed: 09-Jan-2025].
[25] California Department of Transportation (Caltrans), 'Mobile Terrestrial Laser Scanning (MTLS) and Its Applications in Infrastructure Management', California Department of Transportation, 2021.
[26] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, 'End-to-end object detection with transformers', in *European conference on computer vision*, 2020, pp. 213–229.
[27] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, 'Estimating the Support of a High-Dimensional Distribution', *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 07 2001.

[28] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, 'A density-based algorithm for discovering clusters in large spatial databases with noise', in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.
[29] F. T. Liu, K. M. Ting, and Z.-H. Zhou, 'Isolation-Based Anomaly Detection', *ACM Trans. Knowl. Discov. Data*, vol. 6, no. 1, Mar. 2012.
[30] M. A. Fischler and R. C. Bolles, 'Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography', *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[31] S. Suthaharan, 'Support vector machine', *Machine learning models and algorithms for big data classification: thinking with examples for effective learning*, pp. 207–235, 2016.
[32] A. Dosovitskiy *et al.*, 'An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale', in *International Conference on Learning Representations*, 2021.