Enhancing Weed Detection Accuracy Under Tree Crop Canopies Integrating Nano-Drone and UAV Imaging in a Multi-Scale 3D Point Cloud Framework

by

Xizhi Xia

B.A.Sc., University of Tennessee, 2023

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Applied Science

in the
School of Mechatronic Systems Engineering
Faculty of Applied Sciences

© Xizhi Xia 2025
SIMON FRASER UNIVERSITY
Fall 2025

Declaration of Committee

Name: **Xizhi Xia**

Degree: **Master of Applied Science**

Title: **Enhancing Weed Detection Accuracy Under Tree Crop Canopies Integrating Nano-Drone and UAV Imaging in a Multi-Scale 3D Point Cloud Framework**

Committee: **Chair:** **John Shen**
Professor, Mechatronic Systems Engineering

Woo Soo Kim
Supervisor
Professor, Mechatronic Systems Engineering

Bing Lu
Committee Member
Assistant Professor, Geography

Faranak Farzan
Examiner
Professor, Mechatronic Systems Engineering

Abstract

Accurate under-canopy weed detection is a major challenge in precision agriculture due to occlusion in conventional UAV imagery. This study presents an unsupervised 3D point-cloud framework integrating UAV and nano-drone data to improve weed identification in dense crop environments. Field experiments in a blueberry plantation combined UAV (DJI Mavic 3M) and nano-drone (DJI Mini 4 Pro) imagery. Point clouds were co-registered using ground control points, rigid transformation, and Iterative Closest Point (ICP). Segmentation employed Progressive Morphological Filtering (PMF) for ground/non-ground separation, weighted K-means for soil–vegetation classification, and DBSCAN–K-means clustering for individual weed identification. A 2D weed map was generated by projecting crops and weed location. The integrated dataset identified 125 weed clusters compared to 52 with UAV-only data, achieving 72.8% precision, 93.8% recall, and 82.1% F1-score. Results confirm that multi-perspective point clouds enhance under-canopy weed detection and provide a foundation for large-scale weed mapping, and automated weeding systems.

Keywords: UAV mapping; nano-drone; 3D point cloud alignment; unsupervised segmentation; precision agriculture; weed detection

Dedication

I would like to dedicate this thesis to my family and my friends for their unwavering support and encouragement all along.

Acknowledgements

First and foremost, I would like to sincerely thank my supervisor, Dr. Woo Soo Kim, for his constant guidance, encouragement, and patience. His support has not only shaped the direction of this research but also inspired me to grow as a researcher and an individual.

I am deeply grateful to my family, whose unconditional love and support have carried me through every challenge of this journey. Their belief in me has been my greatest source of strength.

I would also like to thank my friends, both in my life and in the Additive Manufacturing Lab, for filling this journey with companionship, encouragement, and countless memorable moments. I am fortunate to have shared this experience with Yiting, Pranjal, Zhengliang, Hasti, Danielle, Sogol, Amir, Atiqa, Hieu, Hadi, and Nancy. Your friendship and support made the hard work lighter and the successes even more meaningful.

Table of Contents

List of Tables

List of Figures

List of Acronyms

3D	Three-dimensional
ANN	Artificial Neural Networks
CNN	Convolutional Neural Networks
CSF	Cloth Simulation Filter
CWSI	Crop Water Stress Index
DBSCAN	Density-based Spatial Clustering of Applications with Noise
DCP	Deep Closest Point
DTM	Digital Terrain Model
ExG	Excess Green Index
ExGR	Excess Green minus Excess Red Index
GCP	Ground Control Point
GMM	Gaussian Mixture Model
GSD	Ground Sampling Distance
ICP	Iterative Closest Point
LiDAR	Light Detection and Ranging
LFSH	Local Feature Statistics Histogram
MOG	Mixture of Gaussians
MVS	Multi-View Stereopsis
NDVI	Normalized Difference Vegetation Index
OSAC	Optimized Sample Consensus
PCL	Point Cloud Library
PMF	Progressive Morphological Filter
RANSAC	Random Sample Consensus
R-CNN	Region-based Convolutional Neural Networks
RF	Random Forest
SfM	Structure from Motion
SMRF	Simple Morphological Filter
SOM	Self-organizing Maps
SVD	Singular Value Decomposition

SVM	Support Vector Machines
UAV	Unmanned Aerial Vehicles
VI	Vegetation Index

Chapter 1. Introduction

1.1. Background

Weeds are undesirable plants that compete with crops for essential resources such as nutrients and water [1], leading to substantial yield reduction and increased production costs. To safeguard agricultural productivity, weed management has long been a major focus of agronomic research. Conventional strategies include physical, mechanical, cultural, biological, and chemical approaches, among which chemical herbicides remain the most widely adopted [2, 3, 4]. However, selective pressure has accelerated the evolution of herbicide resistance in many weed species [5, 6, 7], and excessive herbicide use poses risks to crops, ecosystems, the environment, and human health [8, 9, 10, 11]. These challenges highlight the urgent need for more sustainable and precise weed management strategies [12].

Unmanned Aerial Vehicles (UAVs), commonly referred to as drones, are powered aerial systems that operate without an onboard pilot, relying on aerodynamic forces for lift and functioning either autonomously or under remote control [13]. Equipped with diverse payloads such as RGB, multispectral, hyperspectral, thermal, and Light Detection and Ranging (LiDAR) sensors, UAVs enable the collection of high-resolution imagery and spatial data for environmental monitoring [14]. In practice, UAVs have been successfully deployed across multiple fields, including mining [15], forestry [16], and agriculture [17, 18]. Their adoption enhances the speed of data acquisition, spatial resolution, deployment flexibility, operational safety, and cost-effectiveness compared with traditional manned aircraft or ground-based surveys.

1.1.1. UAV mapping in agricultural use

Unmanned Aerial Vehicles (UAVs) have become one of the most widely adopted platforms for high-resolution mapping. The most basic approach relies on image mosaicking to generate orthophotos; however, more advanced photogrammetric methods such as Structure from Motion (SfM) and Multi-View Stereopsis (MVS) can reconstruct detailed three-dimensional (3D) products, including digital surface and terrain models, textured 3D meshes, and vector-based information layers [19, 20, 21].

The effectiveness of UAV mapping lies in its ability to rapidly survey large agricultural areas with centimeter-level accuracy, while accommodating a wide range of sensor payloads, such as RGB, multispectral, hyperspectral, or LiDAR sensors. These capabilities have established UAVs as indispensable tools for precision agriculture, supporting tasks such as crop growth monitoring [22], yield estimation [23, 24], and weed detection [25]. Beyond spectral analysis, UAV-derived point clouds also provide valuable structural information, enabling advanced analyses such as crop height estimation [26] and canopy volume quantification [27].

Despite these advantages, most UAV mapping workflows are inherently limited by their top-down imaging perspective. In dense crop environments, the canopy obstructs visibility of the under-canopy region, leading to incomplete data collection and the omission of low-lying vegetation. Consequently, features such as under-canopy weeds cannot be reliably detected or mapped using UAV imagery alone. This limitation directly impacts weed management, as undetected weeds contribute to long-term persistence and reduce the effectiveness of control strategies. Overcoming these shortcomings requires mapping methods that complement conventional UAV perspectives with alternative viewpoints, particularly under-canopy imagery, to achieve a more complete 3D reconstruction of the field environment.

1.1.2. Precise weed detection

Most precision weed detection methods including robotic and UAV-based 2D image approaches are based on computer vision and image processing, exploiting feature differences such as texture, shape, spectral reflectance, and color between crops and weeds to achieve classification [28]. The integration of machine learning and deep learning has further improved detection accuracy and automation [29]. Algorithms such as Support Vector Machines (SVM), Artificial Neural Networks (ANN), and Convolutional Neural Networks (CNN) have been widely applied in weed classification tasks [30]. However, reliance on supervised learning models makes these workflows less accessible to end-users, as they often require extensive labeled datasets and frequent model retraining.

Weed management robots represent a typical example of ground-based solutions, where onboard cameras capture close-range imagery for weed recognition [31]. While such systems can achieve high accuracy, their operational coverage is limited, making them less suitable for large-scale farmland applications. UAV-based approaches, by contrast, offer higher scanning efficiency and scalability for field-scale weed detection. Nevertheless, UAV imagery is constrained by flight altitude and its top-down perspective, which restricts visibility under dense canopies and reduces detection reliability in complex field conditions.

In contrast to both robotic and UAV-based image approaches, LiDAR and laser scanning, either equipped on UAV or as ground equipment, methods construct 3D point clouds and apply clustering algorithms to exploit spatial information for weed detection [32, 33]. Although these approaches can outperform traditional UAV imagery in terms of accuracy, they are computationally expensive, require longer processing times, and remain impractical for routine agricultural monitoring.

The limitations of existing approaches highlight the need for new solutions capable of overcoming occlusion challenges while remaining computationally efficient and user-friendly. Unsupervised methods that eliminate the dependency on labeled datasets and training can enhance accessibility, ensure broader applicability, and simplify adoption in real-world farming contexts.

1.1.3. Nano-drones

Nano-drones or nano-UAVs, commonly referring to ultra-light unmanned aerial vehicles under 250 g, are formally defined in aviation safety regulations in several countries, including Canada, Europe, and the United States [34]. Through strict weight budgeting and the removal of non-essential components, nano-drones can achieve dimensions below 10 cm, enabling them to operate effectively in confined or cluttered environments where larger UAVs cannot maneuver [35]. Existing agricultural applications of nano-drones include aerial monitoring inside greenhouses [36] and pollination tasks in controlled environments [37].

Compared with larger UAV platforms, nano-drones can maneuver beneath dense vegetation, fly closer to the ground, and capture understory information that is typically obscured from conventional top-down aerial imagery. However, nano-drones also face inherent limitations, including reduced GPS accuracy, shorter battery life, lower flight stability, and constraints on sensor payload capacity [38]. These limitations prevent nano-drones from replacing conventional UAVs for large-scale field coverage or high-precision mapping when used alone. Nevertheless, their complementary viewing geometry makes them highly valuable for tasks requiring close-range inspection—such as under-canopy weed detection—when deployed alongside a conventional UAV. Integrating data from both platforms enables a more complete 3D reconstruction, combining stable canopy-level structure with detailed ground-level observations and effectively addressing visibility challenges that traditional UAV systems cannot overcome.

1.2. Motivation

Although UAV-based mapping has significantly advanced precision agriculture, its reliance on a top-down imaging perspective restricts visibility in dense crop environments, where canopy cover prevents accurate detection of understorey vegetation. As a result, under-canopy weeds often remain unobserved and untreated, reducing the effectiveness of weed management strategies and contributing to long-term persistence in agricultural fields. To overcome this limitation, there is a need to integrate under-canopy information, such as nano-drone data, with conventional UAV mapping to generate a more complete three-dimensional representation of the field. We define a nano-drone as a smaller drone than a usual UAV, weighing under 250 grams.

At the same time, existing weed detection methods frequently rely on supervised machine learning models, which demand large amounts of labeled data and repeated training to adapt to new crops, environments, or growth stages. These requirements create barriers for practical adoption by farmers, who often lack the time and resources for extensive data preparation. Therefore, this research applies a series of tailored unsupervised algorithms for point cloud alignment and weed detection, ensuring the workflow remains general, efficient, and accessible without the need for labeled datasets. By addressing both visibility and usability challenges, this work aims to provide a scalable and farmer-friendly solution for precise weed mapping.

1.3. Objective and Scope

The objective of this research is to enhance weed detection accuracy by integrating under-canopy nano-drone imagery with conventional high-altitude UAV data to generate a more complete three-dimensional representation of agricultural fields. To achieve this, the study employs a series of unsupervised algorithms for point cloud alignment, crop–ground segmentation, soil–vegetation separation, and individual weed clustering. Techniques such as RGB-weighted K-means, DBSCAN clustering, transformation-based alignment with Iterative Closest Point (ICP), and ground filtering methods are combined to ensure robust and scalable processing without reliance on labeled training data.

The scope of this research includes the development and evaluation of this unsupervised workflow in different agricultural environments, with a focus on under-canopy weed detection and the generation of complete weed distribution maps. The proposed method is compared against conventional UAV-only mapping to demonstrate improvements in accuracy and coverage. While the framework is designed for adaptability across crops and field conditions, extensions such as incorporating multispectral sensors or real-time robotic platforms fall beyond the present scope and are suggested for future investigation.

1.4. Contributions

This thesis introduces a novel framework for under-canopy weed detection in dense crop environments through the integration of nano-drone and UAV imagery. The conceptual idea of combining multi-perspective drone data was suggested by my supervisor, Dr. Woo Soo Kim, while the full algorithmic development was designed and implemented by me. The contributions include point cloud generation, transformation-matrix–based alignment, fine registration using ICP, ground/non-ground separation, soil–vegetation classification via weighted K-means, individual weed segmentation through DBSCAN–K-means clustering, and final weed map creation.

Beyond the core contributions of this thesis, the developed unsupervised algorithm pipeline was extended to related applications, including the extraction of tree height and canopy volume in orchard environments. These studies highlight the broader applicability of the proposed methods for sustainable agricultural monitoring and precision management. The following publication supports the algorithms used in this thesis.:

- X. Xia, J. Sachar, K. Alibhai, J. Labelle, D. Demin, E. Sulle, and W. S. Kim, "Accessible Drone Image Processing for Sustainable Resource Management of 3D Tree-like Crops Using Unsupervised Algorithms," *Information Processing in Agriculture*, https://doi.org/10.1016/j.inpa.2025.11.009, (2025).

1.5. Thesis overview

This thesis aims to improve weed detection accuracy in agricultural fields by integrating under-canopy nano-drone imagery with conventional UAV data through an unsupervised 3D point cloud processing framework. Chapter 2 provides background information and a literature review on UAV-based mapping, weed detection techniques, and point cloud processing methods. Chapter 3 presents the proposed nano-drone assisted workflow, including data acquisition, point cloud alignment, crop and soil segmentation, and unsupervised weed clustering. Chapter 4 discusses the experimental results, highlighting improvements in under-canopy detection and overall weed mapping accuracy compared to conventional UAV only methods. Chapter 5 concludes the research by summarizing the findings and outlining future work, including incorporating weed density assessment and visualization across entire fields, and integrating the generated weed maps with automated devices capable of performing site-specific weed control.

Chapter 2. Literature Review

2.1. UAV mapping in agricultural

UAVs are increasingly employed in agricultural research, with the choice of payload strongly dependent on the monitoring objective. Sensors ranging from simple RGB cameras to advanced multispectral, hyperspectral, thermal, and LiDAR systems enable diverse applications, each providing unique spectral or spatial information.

RGB cameras are the most common payload due to their low cost and high spatial resolution. Studies have shown their effectiveness for tasks such as crop emergence monitoring in potatoes [39], yield estimation in cotton [40], and fruit detection in orchards [41].

Multispectral cameras capture reflectance in discrete bands such as green, red, red-edge, and near-infrared, enabling vegetation index calculations like Normalized Difference Vegetation Index (NDVI) for crop monitoring. Guan et al. demonstrated that UAV-derived NDVI values in rice and wheat fields were strongly correlated with fertilizer application levels and yields, supporting their use in nutrient management and yield prediction [42]. In tree crops, Johansen et al. used UAV multispectral imagery in macadamia orchards to derive vegetation indices that effectively classified tree health conditions, illustrating the value of multispectral payloads for precision agriculture [43].

Hyperspectral sensors further increase spectral resolution, recording hundreds of contiguous bands across the visible and near-infrared spectrum. This enables detection of subtle biochemical variations in crops that multispectral cameras cannot capture. Zarco-Tejada et al. successfully used UAV hyperspectral imagery to estimate leaf carotenoid content in vineyards [44], while Lucieer et al. demonstrated hyperspectral applications for chlorophyll and pigment mapping in heterogeneous agricultural landscapes [45]. These studies confirm the potential of hyperspectral payloads for detailed physiological assessments, though their operational complexity and heavy data requirements remain challenges.

Thermal cameras provide canopy temperature measurements, which can be used as indicators of water stress and irrigation needs. Park et al. used UAV thermal imagery to evaluate crop water status, demonstrating the potential of thermal sensing for precision irrigation management [46]. Such applications illustrate how UAV thermal payloads contribute to water-use efficiency and stress detection in agricultural systems.

LiDAR sensors offer direct acquisition of 3D structural information by measuring distances between the UAV and the vegetation surface. Christiansen et al. demonstrated UAV-mounted LiDAR for agricultural field surveying, with applications in crop height and canopy structure estimation [47]. However, LiDAR systems are often more expensive and data-intensive compared to passive optical sensors. Recent advances in SfM photogrammetry have made UAV RGB imagery a practical alternative for 3D reconstruction, producing dense point clouds that include both geometric and spectral information. For example, UAV RGB point clouds have been applied to canopy volume extraction in citrus orchards [27] and biomass estimation in forage crops [48], showing comparable performance to LiDAR-based approaches but with lower cost and greater accessibility.

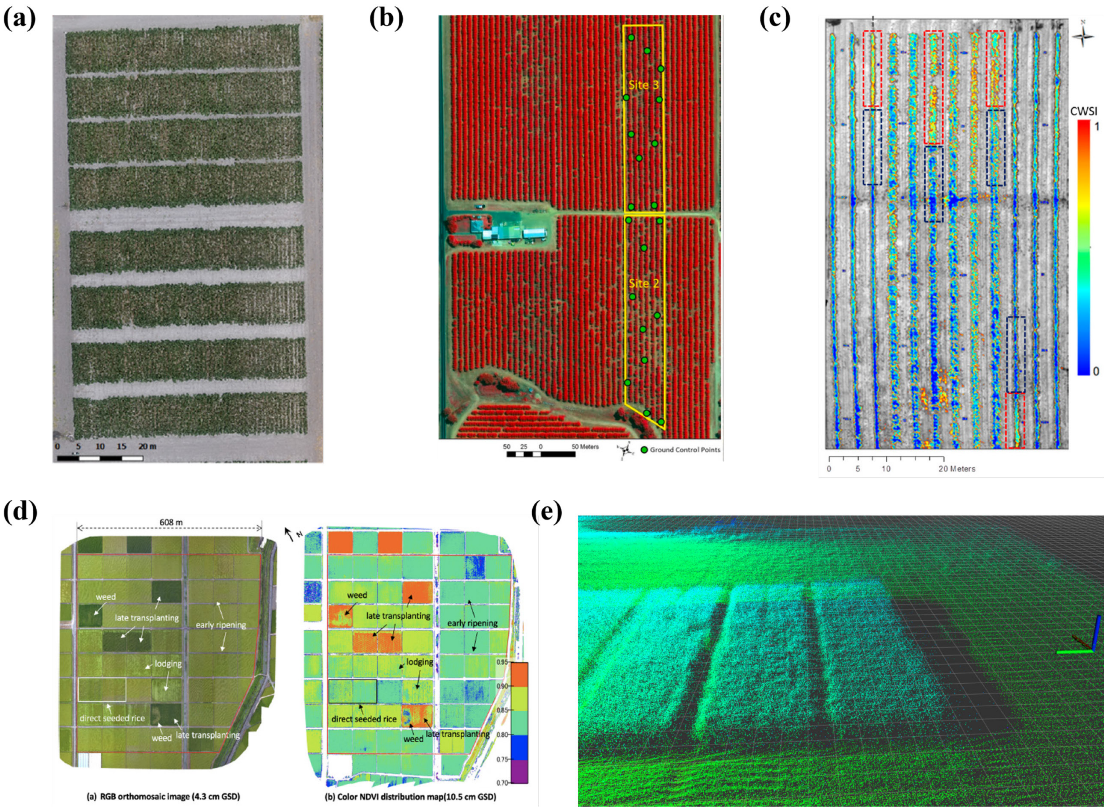

Figure 2.1. Examples of UAV mapping in agricultural applications: (a) UAV RGB orthomosaic of cotton breeding plots [40], (b) UAV false colour orthomosaic of macadamia orchard [43], (c) Adaptive Crop Water Stress Index (CWSI) map derived from UAV thermal infrared image [46], (d) RGB orthomosaic and NDVI distribution map of rice paddies [42], (e) UAV LiDAR point cloud of winter wheat field [47].

In summary, UAV mapping in agriculture leverages different payloads to meet diverse research objectives, from RGB-based crop monitoring to multispectral, hyperspectral, thermal, and LiDAR systems. While each sensor type offers unique advantages, limitations remain, particularly in terms of canopy occlusion and data integration. These challenges highlight the need for innovative workflows that combine complementary perspectives, such as under-canopy nano-drone imagery with conventional UAV mapping, to provide more complete and accurate field representations.

2.2. Weed detection approaches

Weed detection methods can generally be divided into two main categories: those relying on 2D image information and visual analysis, and those based on 3D spatial information combined with clustering algorithms.

2D image-based methods typically exploit differences in spectral and color features, such as RGB (red, green, and blue), HSI (hue, saturation, and intensity), and HSV (hue, saturation, and value) color spaces, to discriminate between crops, weeds, and soil [49]. Multispectral and hyperspectral imaging provide additional spectral information, enabling the calculation of a broader range of vegetation indices (VIs), thereby improving species differentiation [50]. For instance, fusion of RGB and multispectral images has been shown to enhance discrimination of weed patches in rice fields [51], while hyperspectral sensing coupled with machine learning models such as self-organizing maps (SOM), mixture of Gaussians (MOG), and SVM achieved accurate crop–weed discrimination in field studies [52]. The development of machine learning and deep learning further expanded detection accuracy, with methods including SVM [53], Random Forest (RF) [54], ANN [55], CNN [56], YOLO [57], and Region-based Convolutional Neural Networks (R-CNN) [58] applied across various crop field. To reduce the dependence on costly manual labeling, Ferreira et al. developed an unsupervised deep learning framework combined with semi-automatic data labeling, achieving high weed discrimination accuracy while greatly minimizing annotation effort [59]. Table 2.1 provides an overview of widely adopted machine-learning methods for weed detection, highlighting the algorithm type, main features, and performance reported in previous research.

Table 2.1 Review of usual weed detection machine learning algorithms

Research	**ML algorithm**	**Feature**	**Accuracy**
Weed detection in three sunflower fields and two cotton fields [54] *Castro et al. 2017*	Random Forest (RF)	Robust to noise Good with limited data Requires feature engineering	Varies from 59.1% to 84%
Weed detection in carrots crop fields [53] *Murawwat et al. 2018*	Support Vector Machines (SVMs)	Strong for binary classification Sensitive to parameter tuning	Varies from 50% to 95%
Weed detection in sesame and melon fields [55] *Monteiro et al. 2021*	Artificial neural networks (ANNs)	Learns nonlinear patterns Less manual feature design Requires large dataset training	Varies from 92% to 100%
Weed detection in soybean field [56] *Haq 2022*	Convolution Neural Network (CNN)	Automatic feature extraction Requires large dataset training	Grass 98.58%, broadleaf (weed) 98.32%
Weed detection in sesame fields [57] *Chen et al. 2022*	YOLO	Real-time detection Lower accuracy on small weeds	Varies from 83.54% to 90.70%

Research	**ML algorithm**	**Feature**	**Accuracy**
Weed detection in soybean fields [58] *Zhang et al. 2023*	Region-based Convolutional Neural Networks (R-CNN)	High-precision detection Good for small weeds Computationally heavy	Average 99.16%

3D spatial information-based methods leverage plant geometry, height, and volume to overcome some of the limitations of 2D imagery, especially in under-canopy weed detection. Structured-light stereoscopy and depth cameras, have been used to reconstruct 3D plant models, enabling the separation of crops and weeds based on corrected plant height and volumetric parameters [60]. LiDAR and SfM photogrammetry further enhance the availability of point cloud data, but these methods often impose heavier computational costs compared to 2D approaches [61, 62].

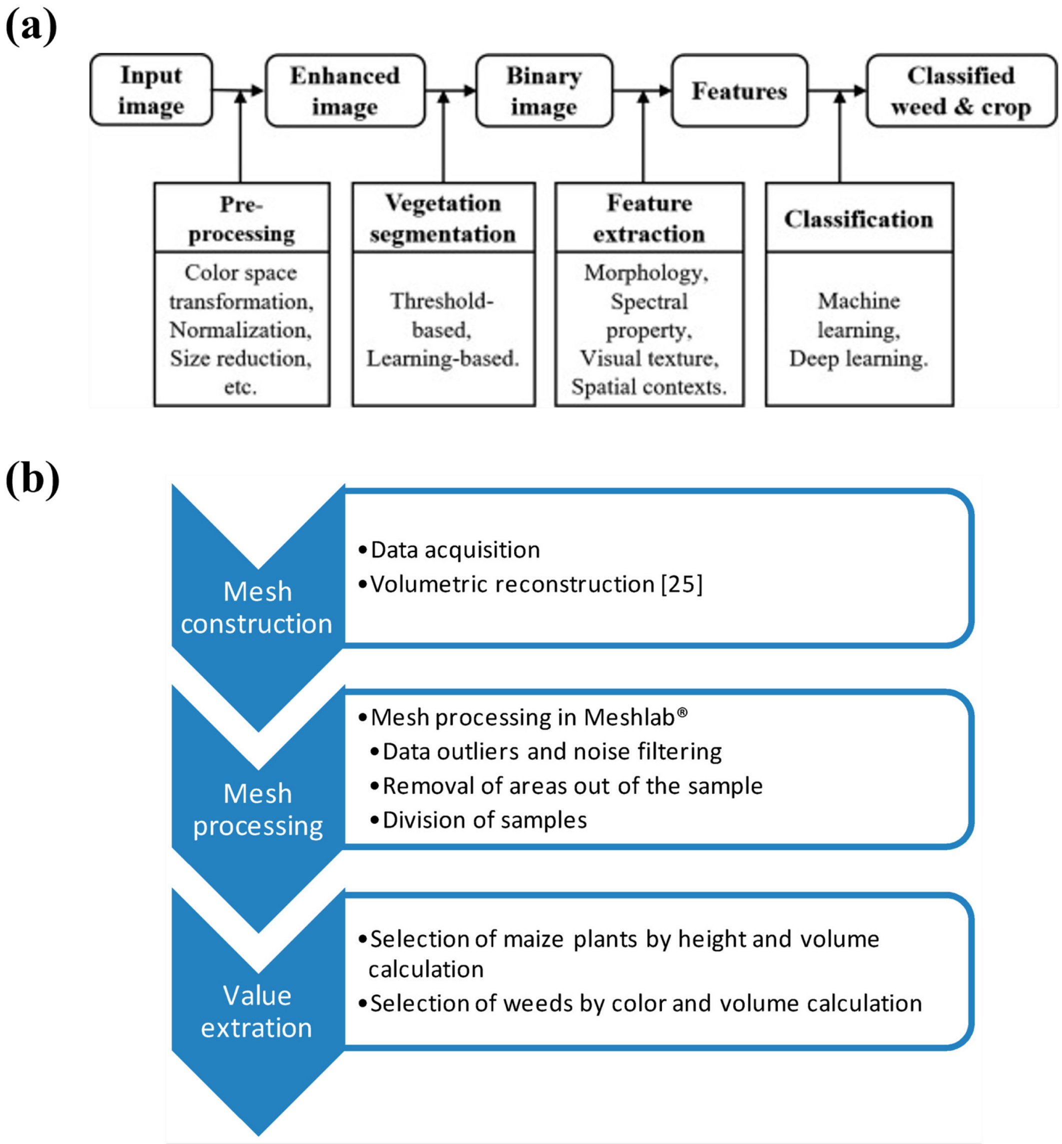
(a)
Input image
Enhanced image
Binary image
Features
Classified weed & crop
Pre-processing
Color space transformation, Normalization, Size reduction, etc.
Vegetation segmentation
Threshold-based, Learning-based.
Feature extraction
Morphology, Spectral property, Visual texture, Spatial contexts.
Classification
Machine learning, Deep learning.
(b)
Mesh construction
•Data acquisition
•Volumetric reconstruction [25]
Mesh processing
•Mesh processing in Meshlab®
•Data outliers and noise filtering
•Removal of areas out of the sample
•Division of samples
Value extration
•Selection of maize plants by height and volume calculation
•Selection of weeds by color and volume calculation

Figure 2.2. Examples of weed detection workflows: (a) a 2D image-based computer vision pipeline, involving preprocessing, vegetation segmentation, feature extraction, and classification [49], (b) a 3D spatial information-based pipeline, incorporating mesh construction, processing, and plant/weed value extraction [60].

In addition, several other studies provide complementary insights. For example, robot-mounted sensor system composed of a photoelectric signal modulation system and a photoelectric signal acquisition and processing system improved the accuracy and efficiency of the spectral reflectance measurement [63]. Likewise, video-based machine vision prototypes tested in potato fields demonstrated real-time weed segmentation and classification using texture and color descriptors with high accuracy under natural light conditions [64].

Overall, 2D image-based methods dominate current practice due to their simplicity and wide adoption, but they are constrained by canopy occlusion and variable illumination. In contrast, 3D approaches offer clear advantages for under-canopy detection and structural analysis, though at the cost of computational efficiency. These complementary strengths underscore the need for developing hybrid workflows that combine UAV and under-canopy imaging with unsupervised algorithms for practical field deployment.

2.3. Photogrammetry and 3D point cloud

Photogrammetry is the science of obtaining reliable spatial information about physical objects and the environment through the recording, measurement, and interpretation of photographic images. Historically, its foundation can be traced back to stereophotogrammetry in the late 19th and early 20th centuries, where overlapping images were used to determine three-dimensional object coordinates [65]. With the development of digital imaging and computational methods, photogrammetry has become one of the most widely used approaches for generating three-dimensional datasets. A key output of modern photogrammetry is the 3D point cloud, which represents objects as discrete sets of points defined by their spatial coordinates (x, y, z) and often associated color or intensity values. Compared to polygon meshes [66], which reconstruct continuous surfaces by connecting vertices into triangles, point clouds are more lightweight and flexible: they directly preserve raw spatial measurements, require less processing for data generation, and are more efficient in terms of storage and compression [67]. With the advancement of laser scanning, LiDAR, and photogrammetry, 3D point clouds have become a standard output format for representing spatial environments [68], and their applications now span construction [69], forestry [70], mining [71], and agriculture [72].

An illustrative example of the value of point clouds in complex environments is found in recent work on under-canopy UAV laser scanning for forestry. Conventional UAV-based surveys conducted above the canopy are limited by occlusion, leading to significant data gaps in densely vegetated forests. In contrast, using UAVs to fly below the canopy with mounted laser scanning systems has been shown to greatly improve the accuracy of tree structural measurements, including diameter, height, and crown geometry [73]. This case highlights the advantage of 3D point clouds in capturing detailed information in challenging, cluttered environments where traditional top-down imaging is insufficient.

2.4. 3D point cloud processing approaches

Processing 3D point cloud data involves a sequence of operations designed to transform raw, unstructured measurements into meaningful geometric and semantic information. The general workflow typically includes filtering, which removes noise and outliers to improve data quality [74]; feature estimation, such as normals and curvatures, that provide geometric descriptors of local structures [75]; surface reconstruction, where point sets are interpolated into continuous surfaces or meshes [76]; and model fitting, where geometric primitives or parametric models are adjusted to approximate observed data [77]. Two additional core tasks are registration, aligning multiple point clouds into a common coordinate system, and segmentation, partitioning the dataset into distinct objects or classes.

To support these tasks, a few specialized software libraries have been developed. Among them, the Point Cloud Library (PCL) is one of the most widely used open-source frameworks, integrating a comprehensive set of algorithms for filtering, feature estimation, registration, segmentation, and surface reconstruction [78]. In addition, point cloud visualization plays a critical role in both qualitative interpretation and quantitative validation. Open3D has emerged as a popular open-source platform offering efficient tools for visualization and processing of 3D data [79].

In the context of this research, registration and segmentation represent the most critical components of the point cloud workflow. Registration ensures accurate alignment between UAV and nano-drone datasets, while segmentation enables the separation of soil, crops, and weeds. Therefore, the following subsections will provide a more detailed literature review of these two processes.

2.4.1. 3D point cloud registration

Registration methods for point clouds can generally be divided into two categories: conventional optimization-based methods and deep neural network-based methods [80]. Within the first category, the most influential is the ICP algorithm, introduced by Besl and

McKay in 1992 [81]. ICP iteratively minimizes the distance between two point sets by alternating between establishing correspondences matching each point in the source cloud with the nearest point in the target and estimating the rigid transformation that minimizes the mean squared error. The process is repeated until convergence, which is guaranteed to a local minimum, though performance depends heavily on the initial alignment.

Besides ICP, Procrustes analysis [82] has also been applied for point cloud alignment. The classical Procrustes method seeks the optimal rigid transformation (rotation, translation, and optionally scale) between two point sets by minimizing least-squares residuals. Building on this foundation, Toldo et al. proposed a global registration approach that embeds Generalized Procrustes Analysis into an ICP framework, enabling simultaneous alignment of multiple views while distributing registration errors evenly across all datasets [83]. This hybrid strategy leverages the robustness of ICP in handling partial overlaps with the mathematical rigor of Procrustes optimization, thus mitigating the accumulation of errors that typically arise in sequential pairwise registration.

Beyond these foundational methods, significant efforts have been made to improve the robustness and efficiency of registration, particularly under conditions of noise, varying density, and partial overlap. For example, Yang et al. proposed a local feature statistics histogram (LFSH) descriptor combined with an optimized sample consensus (OSAC) algorithm, forming a coarse-to-fine pipeline that significantly improved efficiency and robustness in both model and scene registration tasks [84]. Huang et al. introduced a method based on Gaussian Mixture Models (GMMs) which focus more on global information while ignore local structure distortion [85].

In addition to optimization-based techniques, a growing body of research has applied deep neural networks to point cloud registration. PointNetLK adapts the Lucas–Kanade framework by leveraging PointNet features as an alignment function, eliminating explicit correspondence search. Deep Closest Point (DCP) further integrates attention mechanisms and a differentiable Singular Value Decomposition (SVD) layer, allowing end-to-end rigid transformation estimation with higher accuracy under large motions. CorsNet concatenates the local features with the global features and regresses the point cloud correspondence [86, 87, 88]. Collectively, these methods highlight the potential of learning-based approaches to improve robustness to noise, initialization, and cross-source variation.

Overall, conventional optimization-based methods remain attractive for their interpretability, computational efficiency, and solid mathematical guarantees. In contrast, deep learning–based methods demonstrate stronger adaptability to complex, large-scale, or cross-source data by learning robust feature representations directly from data. Together, these two approaches are complementary: optimization-based methods provide reliable baselines with low computational cost, while deep learning approaches offer enhanced generalization and robustness, particularly in challenging field environments.

2.4.2. 3D point cloud segmentation

Segmentation is the process of grouping raw point clouds into subsets that share common geometric or radiometric properties, such as surface orientation, curvature, or reflectance. It is an essential step in point cloud processing, as it enables subsequent classification, modeling, and analysis by reducing complexity and giving structure to otherwise unorganized data.

Conventional segmentation methods can be broadly categorized into several groups. Edge-based segmentation relies on detecting discontinuities in local surface properties such as normals or curvature to delineate object boundaries but often suffers from sensitivity to noise. Region growing methods start from seed points and expand clusters based on similarity criteria like curvature or planarity, offering more robustness to noise but requiring careful parameter tuning. Model-fitting approaches, such as Random Sample Consensus (RANSAC) or Hough Transform, extract geometric primitives (planes, spheres, cylinders) directly from point sets and are particularly effective for man-made structures. Hybrid methods combine multiple strategies to leverage the strengths of different approaches [89].

With the advancement of machine learning, segmentation methods have expanded into supervised, unsupervised, and semi-supervised approaches. Supervised methods rely on annotated datasets to learn semantic classes. Widely used models include RF, Markov Network, SVM, VoxNet, PointNet, ACNN [90]. Zhou et al. ulitized SVM and boosting to segment soybean plants from 3D point cloud [91], and Qi et al. verified segmentation performance of PointNet++, MinkowskiNet and FPconv in the research [27].

Unsupervised methods partition point clouds without labels, leveraging intrinsic data properties. The most common examples are K-means and Density-based spatial clustering of applications with noise (DBSCAN). K-means, first introduced by MacQueen in 1967, partitions data into k clusters by minimizing within-cluster variance [92, 93]. While effective, it requires the number of clusters to be predefined. To address this limitation, Sinaga and Yang proposed an unsupervised K-means that automatically estimates the optimal cluster number without initialization [94]. DBSCAN, on the other hand, groups points based on density, marking sparse points as noise [95]. It is widely used for irregular and non-spherical clusters but is sensitive to parameters. Building upon this, Border-Peeling Clustering was introduced as a non-parametric extension that iteratively peels away border points to reveal cluster cores, thereby adapting more flexibly to varying densities [96].

Semi-supervised methods attempt to reduce annotation cost by combining a small set of labeled data with unsupervised clustering. A representative approach is CANUPO, which applies a multi-scale dimensionality criterion to lidar point clouds for classifying natural scenes with limited supervision, showing utility in geomorphology and other complex environments [97].

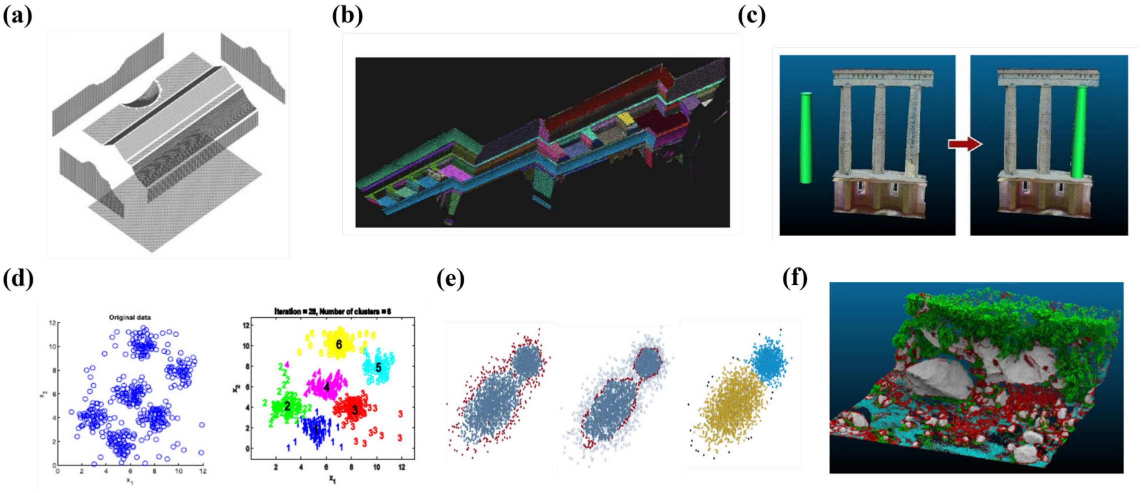

Figure 2.3. Examples of 3D point cloud segmentation result by: (a) edge-based segmentation [98], (b) region growing method [89], (c) model-fitting approach [89], (d) unsupervised K-means clustering [94], (e) border-peeling clustering [96], (f) semi-supervised method: CANUPO [97].

Supervised methods rely on annotated training datasets to learn semantic classes, unsupervised methods automatically partition data based on intrinsic properties without labels, while semi-supervised methods combine limited annotations with unsupervised clustering to reduce labeling effort. These approaches enable higher flexibility and adaptability to diverse datasets, though they may also require significant computational resources.

A Digital Terrain Model (DTM) is an ordered array of numbers representing the spatial distribution of terrain characteristics [99]. It provides a bare-earth representation by removing above-ground features such as vegetation and buildings. DTMs are essential in diverse applications, including flood modeling [100], landslide prediction [101], hydrological analysis [102], forestry inventory [103], and precision agriculture [26], where accurate knowledge of the terrain surface forms the foundation for reliable decision-making.

DTM extraction is therefore a key objective in point cloud segmentation. Among the commonly used geometric filters, the Progressive Morphological Filter (PMF) gradually increases the window size of morphological operators combined with elevation difference thresholds to remove nonground features while retaining ground points [104]. This approach has shown good performance in both flat and mountainous terrains, though it may produce omission errors in steep areas where true terrain points are incorrectly removed. Another widely used method is the Cloth Simulation Filter (CSF), which treats the inverted point cloud as a virtual surface and simulates a cloth dropping over it [105]. The cloth surface adapts to the ground while ignoring elevated objects, making CSF intuitive and effective across various landscapes. However, its accuracy can be influenced by parameter settings such as grid resolution and cloth stiffness.

Beyond these, other approaches have been developed. The Grid-Based Approach rasterizes the point cloud into regular grid cells, then applies hierarchical filtering and weighting functions to detect and replace nonground elements. This method leverages efficient image-processing techniques and allows high-resolution DTMs to be generated from airborne laser scanning data [106]. Similarly, the Simple Morphological Filter (SMRF) modifies the classic morphological filtering framework with adaptive elevation thresholds, offering an alternative for separating ground from nonground points in complex environments [107].

Together, these methods provide the foundation for generating accurate DTMs from UAV or airborne LiDAR point clouds, balancing trade-offs between computational efficiency, adaptability to terrain types, and sensitivity to parameter selection.

Chapter 3. Nano-drone Assisted Unsupervised Weed Detection

3.1. System workflow

The proposed nano-drone assisted weed detection framework consists of four interconnected modules, as illustrated in Figure 3.1.

The first module is Data Acquisition, where imagery is collected using both conventional UAV and nano-drone. This dual-perspective acquisition ensures that information from both canopy and sub-canopy layers is captured, overcoming the occlusion problem of traditional UAV-based mapping.

The second module is Point Cloud Processing and Alignment, where imagery from both sources is processed into 3D point clouds. Reference points are extracted to support the initial alignment through a transformation matrix, followed by fine alignment using the Iterative Closest Point (ICP) algorithm. This ensures accurate registration between multi-source datasets.

The third module, Crop Segmentation, partitions the integrated point cloud into ground and non-ground points. From this process, the non-ground subset is isolated to represent crop structures, providing the foundation for subsequent vegetation analysis.

The final module is the Weed Classifier, which applies unsupervised clustering algorithms to the ground point cloud. Soil–vegetation segmentation is first performed using a weighted K-means algorithm, followed by individual weed segmentation with clustering methods. The outputs are then compiled into a weed distribution map, enabling clear visualization of weed presence and patterns across the field.

This modular workflow integrates multi-perspective data acquisition with unsupervised point cloud processing, allowing for precise, scalable, and training-free weed mapping suitable for diverse agricultural environments.

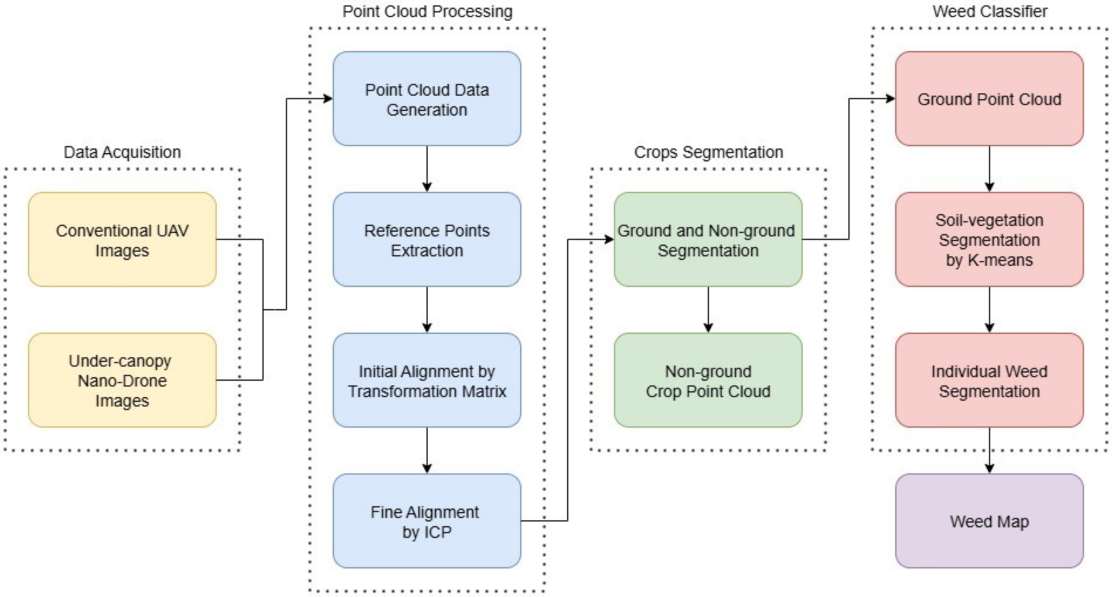
Data Acquisition
Conventional UAV Images
Under-canopy Nano-Drone Images
Point Cloud Processing
Point Cloud Data Generation
Reference Points Extraction
Initial Alignment by Transformation Matrix
Fine Alignment by ICP
Crops Segmentation
Ground and Non-ground Segmentation
Non-ground Crop Point Cloud
Weed Classifier
Ground Point Cloud
Soil-vegetation Segmentation by K-means
Individual Weed Segmentation
Weed Map

Figure 3.1. Workflow chart of the nano-drone assisted weed detection

3.2. Data acquisition

3.2.1. Experiment site

The experiment was conducted in a blueberry field located in Abbotsford, British Columbia, Canada (49.02°N, 122.43°W). The study area covers approximately 675 m² and consists of five rows of blueberry crops, with each row containing 25–35 individual plants. This site was selected as a representative small-scale field, and the planting density in this field is relatively high, leading to significant canopy closure. Under such conditions, conventional UAVs with a top-down perspective often fail to capture under-canopy details due to occlusion from dense foliage. This limitation highlights the need for integrating nano-drone imaging with conventional UAV mapping to obtain more complete 3D information for crop and weed analysis.

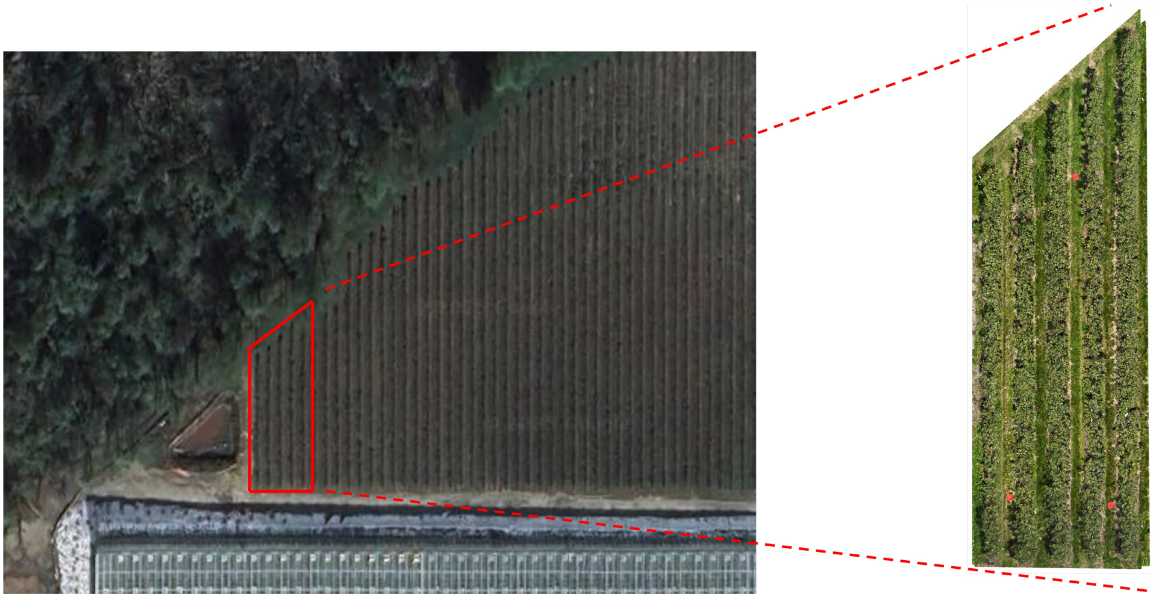

Figure 3.2. Satellite image of the experiment site in Abbotsford, BC, Canada, and the zoomed section showing densely planted blueberry rows.

Figure 3.2 presents the satellite image of the experimental site along with a zoomed-in section of the crop rows, illustrating the dense planting structure and potential under-canopy occlusion.

3.2.2. UAV and nano-drone image acquisition

Two UAV platforms were employed in this study: the DJI Mavic 3M (SZ DJI Technology Co. Ltd, Shenzhen, China), referred to as the conventional UAV, and the DJI Mini 4 Pro (SZ DJI Technology Co. Ltd, Shenzhen, China), used as the nano-drone. The Mavic 3M is equipped with a 5280 × 3956 pixel camera and was flown at an altitude of 150 ft (45.72 m) to capture nadir images of the experimental field. In contrast, the Mini 4 Pro provides a higher resolution of 8064 × 6048 pixels and was deployed at a much lower altitude of 6 m above its departure location, enabling detailed imaging of under-canopy regions.

Ground Control Points (GCPs) were placed across the site to improve point cloud accuracy and facilitate multi-source alignment [108, 109]. Four standard chessboard GCPs were positioned at the four corners of the field to enhance photogrammetric reconstruction precision. Additionally, three custom red GCPs were distributed in a triangular configuration within the field to serve as reference points for the transformation matrix during alignment. These GCPs were fabricated from polyester fabric, which minimized light reflection variability under changing illumination conditions and improved image consistency. Photographs of the UAV platforms and GCPs are shown in Figure 3.3.

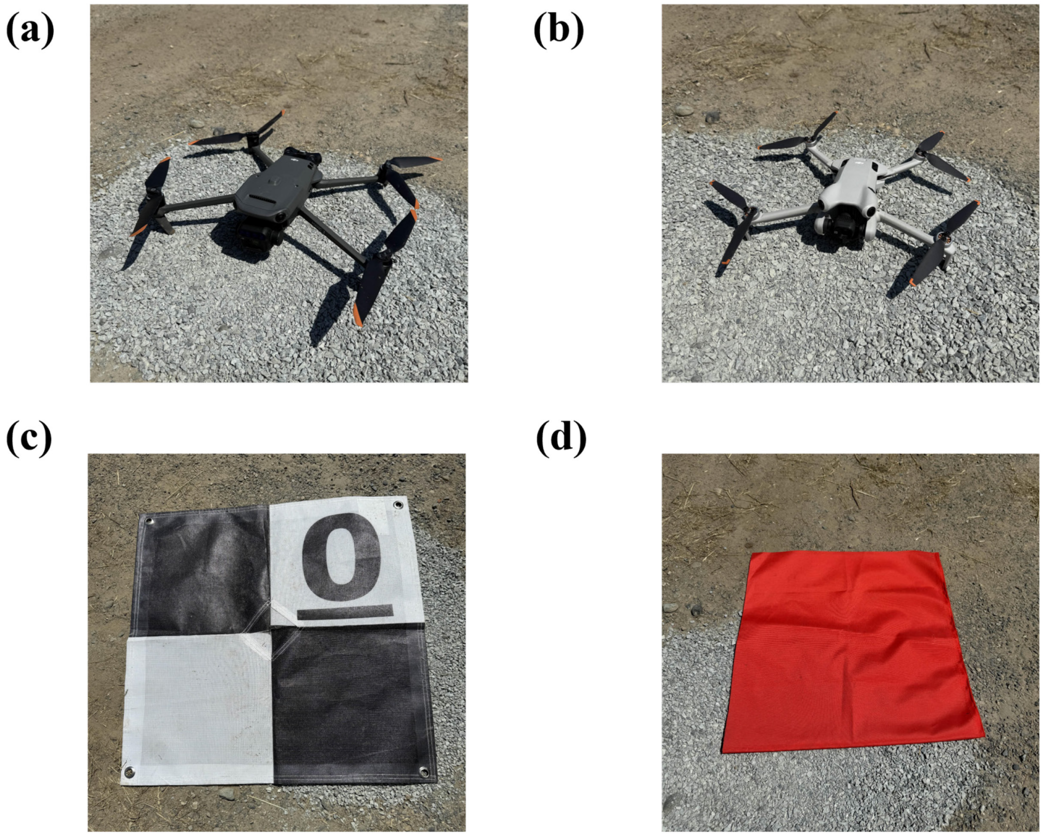

Figure 3.3. UAV platforms and Ground Control Points (a) DJI Mavic 3M, (b) DJI Mini 4 Pro, (c) chess-pattern GCP, (d) custom red GCP.

The Mavic 3M flight path was automatically generated by DJI Smart Farm software, following a structured grid pattern with two orthogonal directions to maximize coverage (Figure 3.4a). In contrast, the Mini 4 Pro required manually defined waypoints due to the low altitude of flight near the canopy, where precise maneuvering was essential. Given its lightweight design (249 g), the nano-drone was more susceptible to instability during flight, resulting in a less regular distribution of capture points (Figure 3.4b). For both UAVs, the image acquisition settings included an 80% frontal overlap and 75% side overlap. The image interval of the nano-drone was calculated based on its Ground Sampling Distance (GSD) and flight altitude.

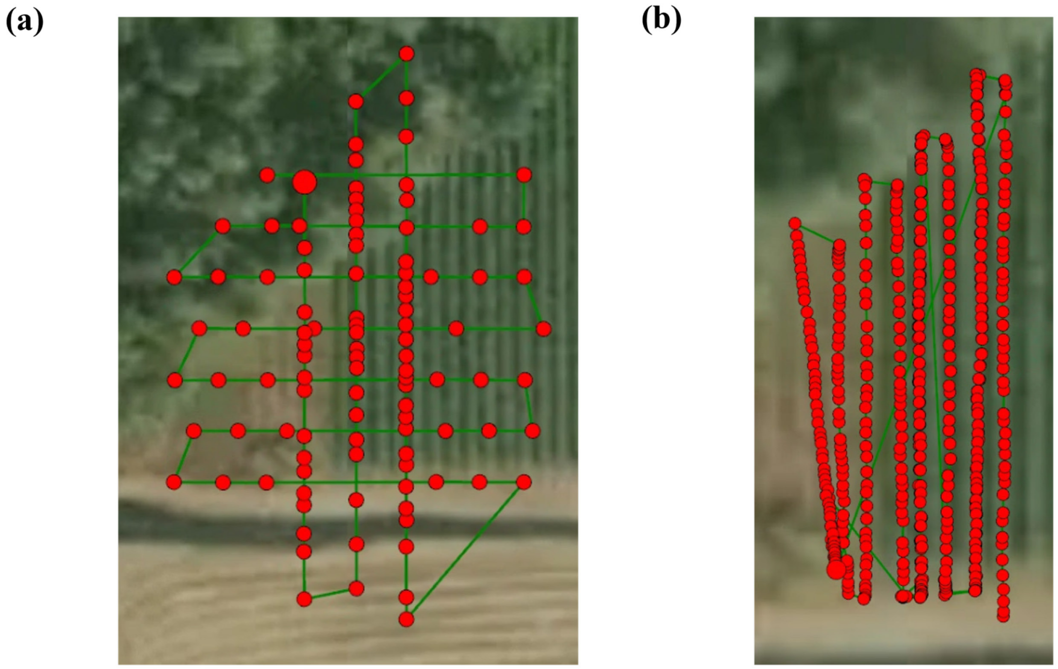

Figure 3.4. UAV flight paths (a) automatic grid flight plan of DJI Mavic 3M, (b) manual setup waypoints path of DJI Mini 4 Pro.

3.2.3. Ground truth

On the same day as the UAV and nano-drone data acquisition, a manual ground truth survey of weed distribution was conducted. The dominant weed species observed in the blueberry field included Curly Dock, Brome Grass, Houttuynia cordata, Thistle, and Mouse-ear Chickweed, with representative field photographs shown in Figure 3.5. During this process, small blueberry offshoots that grew close to the ground and visually resembled weeds were carefully excluded to ensure accuracy.

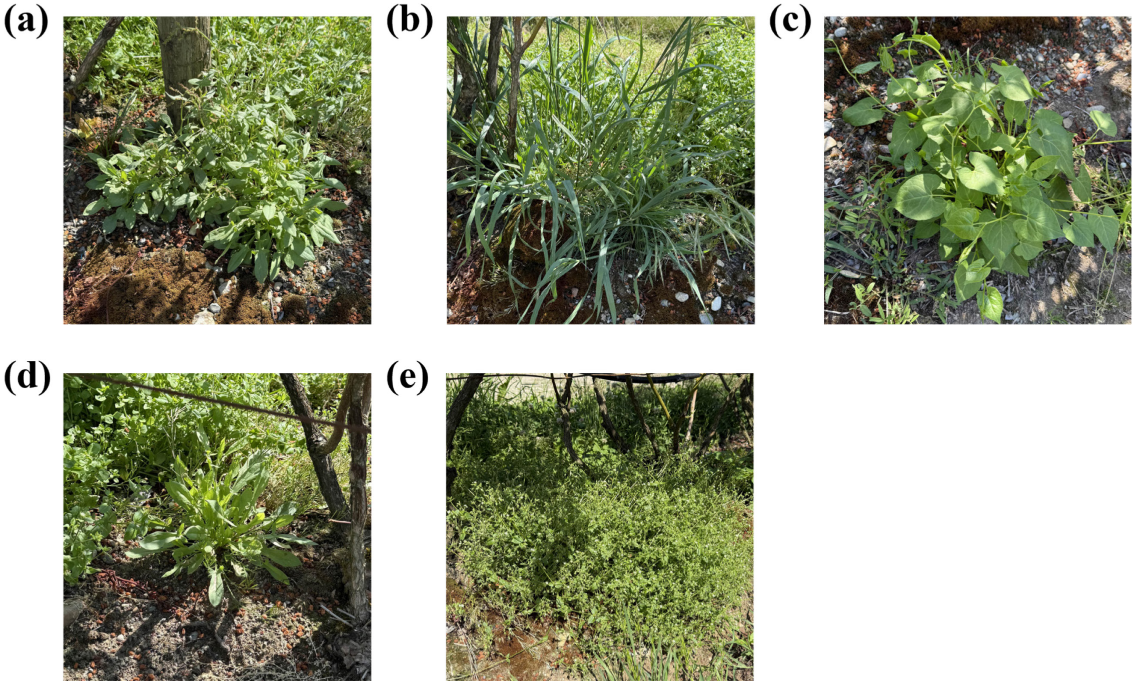

Figure 3.5. Field photographs of representative weed species in the blueberry experimental site: (a) Curly Dock, (b) Brome Grass, (c) Houttuynia cordata, (d) Thistle, (e) Mouse-ear Chickweed.

Due to the dense planting density of the blueberry rows, most weeds were located beneath the canopy and could not be captured by conventional nadir UAV imagery. This limitation highlighted the necessity of adopting a nano-drone-assisted UAV mapping and 3D point cloud-based weed detection strategy. A total of 97 weeds were recorded: 73 under the canopy and 24 in inter-row gaps or alongside blueberry plants. The ground truth data were subsequently digitized into a weed distribution map, which served as a reference baseline for evaluating the performance of the proposed detection algorithm.

3.3. Point cloud data processing and alignment

Following the acquisition of UAV and nano-drone imagery, point cloud data were generated through photogrammetric reconstruction. However, due to differences in sensor type, flight altitude, and imaging perspective, the resulting point clouds often exhibit discrepancies in scale, orientation, and completeness. To ensure that both datasets could be effectively integrated, a systematic point cloud processing and alignment pipeline was implemented. The process involved reference point extraction, initial alignment using transformation matrices, and refinement through iterative optimization methods. This step is critical for fusing complementary top-down and under-canopy perspectives, thereby producing a unified and structurally consistent 3D representation of the crop field.

Conventional UAVs provide a broad top-down perspective, enabling efficient reconstruction of canopy structures across large areas with consistent coverage. Their main limitation lies in the lack of under-canopy visibility, where weeds and stems are heavily occluded. In contrast, nano-drones flown at low altitudes offer valuable under-canopy details, capturing structural information that UAVs cannot access. However, nano-drone scanning also presents challenges, including lower GPS accuracy, reduced flight stability due to their lightweight design, and limited coverage per flight. By integrating these two complementary perspectives, it becomes possible to generate a unified 3D model that preserves canopy integrity while enhancing understorey representation. Figure 3.6 provides a conceptual illustration of this integration process, highlighting how UAV and nano-drone data collectively contribute to more complete field reconstructions.

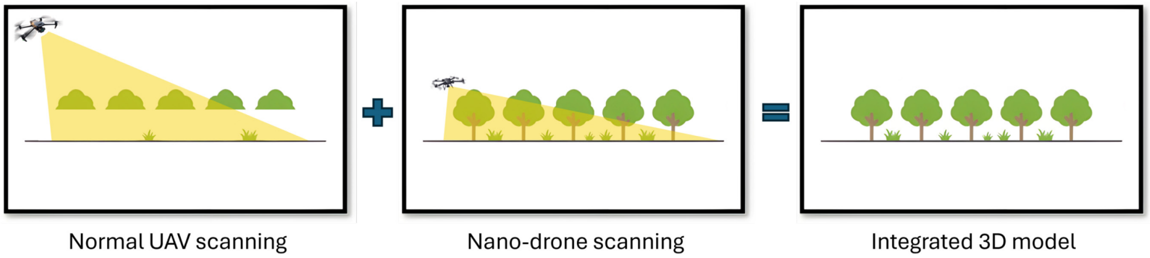

Figure 3.6. Conceptual illustration of UAV and nano-drone complementary scanning for integrated 3D reconstruction.

An illustrative example is presented in Figure 3.6, showing point clouds collected in Central Park, Burnaby, BC, Canada (49.22°N, 123.01°W). The conventional UAV-derived point cloud provides detailed canopy structure but lacks under-canopy visibility, while the nano-drone-derived point cloud captures understory details at the cost of reduced coverage. Their alignment and fusion demonstrate the potential to generate more complete reconstructions by integrating multi-perspective data.

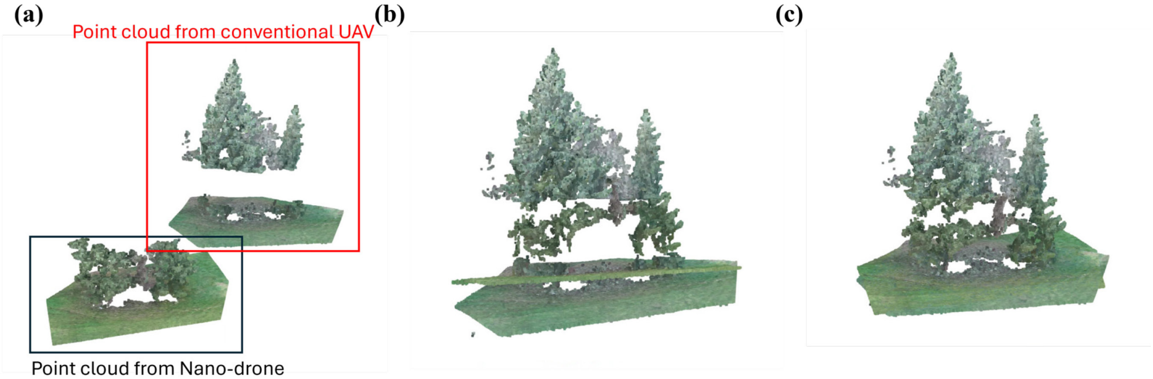
(a)
Point cloud from conventional UAV
Point cloud from Nano-drone
(b)
(c)

Figure 3.7. Example of point cloud alignment by steps (a) raw point cloud by two sources, (b) integrated point cloud by transformation matrix, (c) integrated point cloud by ICP fine alignment.

3.3.1. Point cloud data generation and processing

All UAV and nano-drone images were processed in Pix4Dmapper (Pix4D S.A., Prilly, Switzerland) to generate 3D point clouds through standard photogrammetric workflows, including feature matching, bundle adjustment, and dense reconstruction. The resulting point cloud datasets were exported in PLY format, with spatial information stored in an arbitrary coordinate system using metric units (meters). RGB color attributes were normalized to the range of 0–1, facilitating efficient reading and visualization in Python using the Open3D library.

Since the visualization capacity of Open3D is limited by window size and rendering quality, CloudCompare [110] was additionally employed to present high-resolution visualizations and to examine finer details of the reconstructed structures.

Denoising is also an important step in point cloud processing, as it improves the overall quality of the dataset by removing outliers. In this study, denoising was carried out using two types of filters to improve point cloud quality by removing outliers. The statistical outlier removal filter identifies and eliminates points whose average distance to a fixed number of neighbors exceeds a threshold defined by the standard deviation [111]. This approach preserves the overall structure of the dataset while filtering away sparsely distributed noise. Complementing this, the radius outlier filter evaluates the local point density within a specified radius and removes any point that has fewer than a minimum number of neighbors [112]. This method is particularly effective at suppressing small spurious clusters, such as isolated artifacts near the ground surface. Together, these two filters ensured a cleaner and more reliable dataset for subsequent processing steps.

3.3.2. Initial Alignment by Transformation Matrix

The purpose of this step was to achieve a coarse registration of the UAV and nano-drone point clouds by constructing a transformation matrix from three common reference points. Compared with applying ICP directly, this procedure reduces computational load and provides a more stable initialization.

The reference points were obtained from the Red Ground Control Points (GCPs) placed in the field. Although the theoretical RGB value of these GCPs is (255, 0, 0), the actual field conditions—such as illumination and surface reflectance—prevent them from appearing as perfectly pure red. To automate their detection, a DBSCAN clustering algorithm with RGB thresholds was applied. Only points with red values greater than 180, green values less than 120, and blue values less than 100 were retained for clustering, thresholds that were empirically determined from the observed GCP colors. Because noise points in the scene could still satisfy these conditions, a relatively high *min_samples* parameter was used, ensuring that only dense red clusters corresponding to the GCPs were preserved.

For each detected GCP, the centroid coordinate of the cluster was extracted to represent its spatial location. This yielded three points from the UAV point cloud and three from the nano-drone point cloud. To establish one-to-one correspondence between them, a triangle geometry–based ordering method was used. The procedure compared the side lengths of triangles formed by the three GCPs in each dataset and tested all permutations to identify the configuration that minimized the difference in side length distributions. In this way, the two sets of reference points were matched consistently in space.

Finally, with three corresponding pairs of reference points, a rigid transformation matrix consisting of rotation and translation was computed. This transformation completed the initial alignment of the nano-drone point cloud to the UAV point cloud, providing a well-integrated dataset for subsequent processing.

3.3.3. Fine Alignment by ICP

After the initial alignment was established through the transformation matrix, the Iterative Closest Point (ICP) algorithm was employed to refine the registration between the UAV and nano-drone point clouds. ICP operates by iteratively minimizing the distance between corresponding points in two datasets. In each iteration, correspondences are first identified by pairing each point in the source cloud with its nearest neighbor in the target cloud. A rigid transformation consisting of rotation and translation is then estimated to minimize the mean squared error of these correspondences. This process is repeated until convergence criteria are met.

The performance of ICP depends largely on its parameter settings. Key parameters include the maximum correspondence distance, which restricts the search radius for matching points, and the convergence tolerance, which defines the threshold for stopping iterations once the alignment error stabilizes. In this study, these parameters were tuned to balance computational efficiency with registration accuracy, ensuring reliable alignment despite differences in density and perspective between the two point clouds.

Upon completion of the ICP refinement, the UAV- and nano-drone–derived datasets were merged into a single, well-integrated point cloud. This integrated dataset provided both canopy-level and under-canopy information, forming the foundation for subsequent weed segmentation and mapping.

3.4. Crop Segmentation

The first step toward weed detection is to isolate crops from ground points, which includes soil, weed and other non-weed vegetation, in the reconstructed 3D point cloud. Separating crop structures from the underlying terrain ensures that subsequent segmentation and classification processes focus only on relevant plant information while reducing computational complexity.

In this workflow, the Progressive Morphological Filter (PMF) was applied for ground and non-ground separation. The advantage of PMF lies in its robustness when dealing with point clouds that are not perfectly co-registered. Unlike the Cloth Simulation Filter (CSF), which relies on a simulated physical surface to approximate the terrain, PMF incrementally expands a structuring element to detect and remove ground points based on local elevation differences. This morphological approach is less sensitive to misalignments between datasets, making it particularly suitable for our fused UAV and nano-drone point cloud.

During experiments, we observed that when applied to point clouds generated solely from conventional UAV data, both CSF and PMF yielded comparable results. However, when processing the combined multi-perspective dataset, PMF consistently performed better, producing cleaner ground segmentation and more reliable extraction of crop structures. This advantage provided a more accurate foundation for the subsequent soil–vegetation and weed segmentation steps.

3.5. Weed Classifier

3.5.1. Soil-vegetation Segmentation

After crop segmentation, the remaining ground points were further classified into two categories: soil and vegetation (including weeds and small non-crop plants). This soil–vegetation separation was performed using a weighted K-means clustering algorithm, where six features were selected as input: *x*, *y*, relative *z*, Excess Green Index (ExG), and Excess Green minus Excess Red Index (ExGR). The feature weights were set as [0.4, 0.4, 2.3, 2.0, 1.0], respectively, to emphasize the most discriminative attributes.

Relative elevation (*relative z*) was chosen instead of the absolute *z* value because the point clouds were reconstructed in an arbitrary coordinate system. By normalizing elevation relative to local ground level, the feature more reliably distinguished vegetation protruding above the soil surface from background terrain irregularities.

The ExG and ExGR indices were derived from RGB color attributes to capture spectral differences between vegetation and soil. ExG enhances the contribution of the green channel relative to red and blue, making it sensitive to chlorophyll-rich plant structures. ExGR further contrasts green dominance against red dominance, improving robustness under variable illumination. Strengthening the weights of relative *z*, ExG, and ExGR ensured that clustering was driven primarily by structural height and vegetation-specific spectral properties, which are the most reliable indicators for separating soil from vegetation in complex agricultural environments.

The resulting vegetation point cloud was retained as the input for the next stage of processing, namely individual weed segmentation. By filtering out soil background and preserving only vegetation-related structures, the workflow ensured that subsequent clustering focused directly on detecting and isolating individual weed instances from the crop field.

3.5.2. Individual Weed Segmentation

Prior to individual weed segmentation, the vegetation point cloud was denoised to remove spurious points and residual crop tips that might remain after DTM extraction. This preprocessing ensured that subsequent clustering would focus on meaningful vegetation structures rather than noise artifacts.

The cleaned vegetation cloud was then segmented using the DBSCAN algorithm, which partitions data into clusters of varying density without requiring prior knowledge of the number of groups. The resulting clusters represented a mixture of grass patches, individual weeds, and occasional noise components. For each cluster, several geometric features were computed, including maximum linear extent, two-dimensional projected area, and three-dimensional volume. These descriptors provided quantitative measures of cluster morphology that could help distinguish weeds from other vegetation.

Using these features, a K-means clustering algorithm was applied to divide all clusters into two categories: weed clusters and non-weed clusters. Field observations revealed that the maximum length of common weeds in the experimental site rarely exceeded 80 cm. Based on this observation, a post-processing filter was introduced to discard clusters with maximum lengths above this threshold, thereby improving classification accuracy by excluding unlikely candidates.

This combination of density-based clustering, geometric feature extraction, and rule-based filtering produced a reliable weed segmentation outcome, forming the basis for constructing the final weed distribution map.

3.5.3. Weed map creation

Following the identification of weed clusters, the final step was to generate a 2D visualization of weed distribution across the field. Both the crop point cloud and the weed point cloud were projected onto a two-dimensional plane. To enhance interpretability, the crop projection was rendered in green, while the weed projection was rendered in red. The weed layer was then overlaid on top of the crop background, producing a weed distribution map that provides an intuitive view of weed locations relative to crop rows. This map not only facilitates visual assessment of weed density and spatial distribution but also serves as a practical output format for future integration with precision agriculture tools.

Chapter 4. Results and Discussion

4.1. Point Cloud Data

The experiment produced three sets of point clouds: one derived from conventional UAV imagery, one from nano-drone imagery, and a final integrated dataset combining both sources (Figure 4.1). Each dataset exhibited distinct structural and visual characteristics that directly reflect their acquisition perspectives.

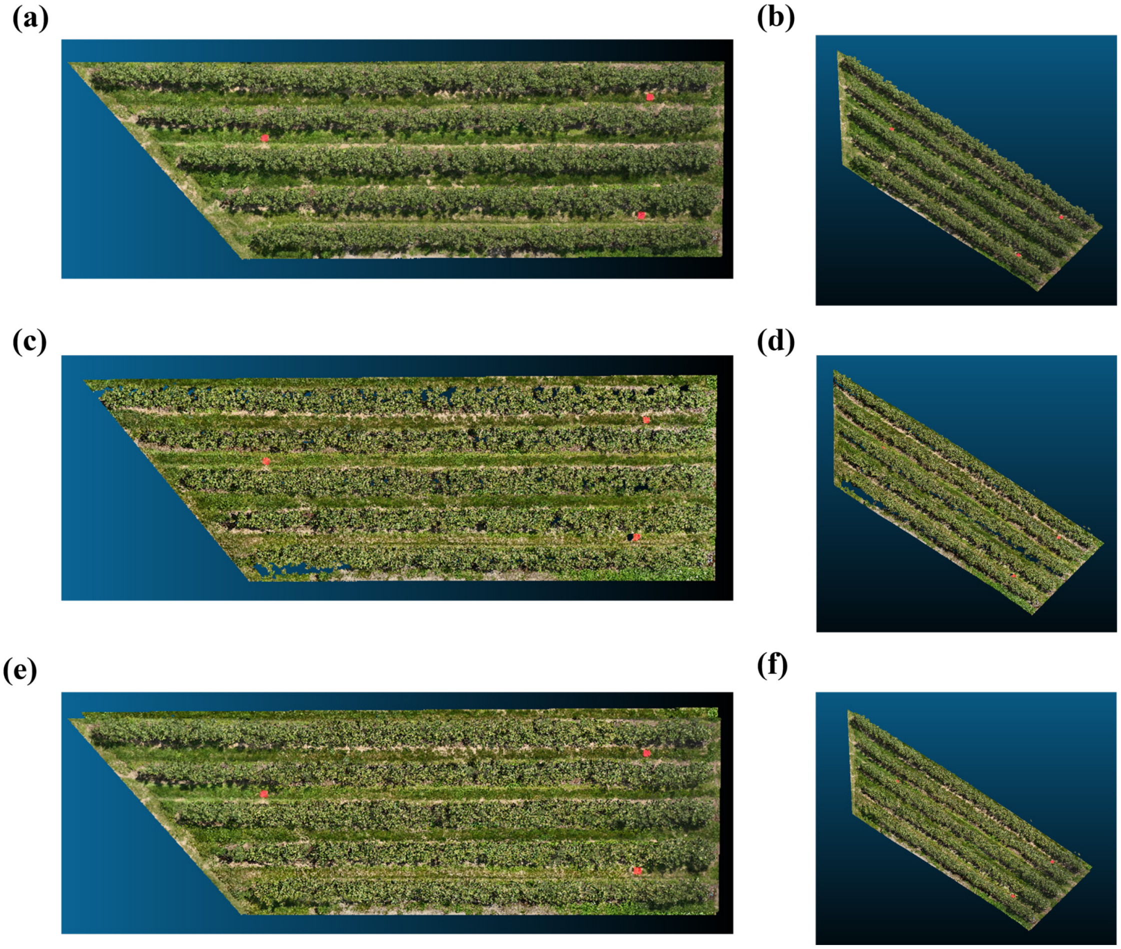

Figure 4.1. Visualization of the reconstructed point clouds at the experimental blueberry field: (a) orthographic view of conventional UAV-derived point cloud, (b) oblique view of conventional UAV-derived point cloud, (c) orthographic view of nano-drone–derived point cloud, (d) oblique view of nano-drone–derived point cloud, (e) orthographic view of the integrated point cloud, (f) oblique view of the integrated point cloud.

The conventional UAV point cloud provided a continuous top-down reconstruction of the crop canopy. While the canopy rows were well defined, the under-canopy regions were largely occluded, leaving visible gaps beneath the blueberry shrubs. In contrast, the nano-drone point cloud, captured from a near-ground perspective, successfully represented under-canopy structures, including stems and low-lying weeds. However, due to close-range flights and the limited stability of the lightweight platform, the nano-drone data appeared less uniform, with redundancy and irregular point spacing. The integrated dataset combined the complementary strengths of both sources, producing a more complete reconstruction that included both canopy-level and under-canopy details, thereby providing a stronger foundation for subsequent weed segmentation.

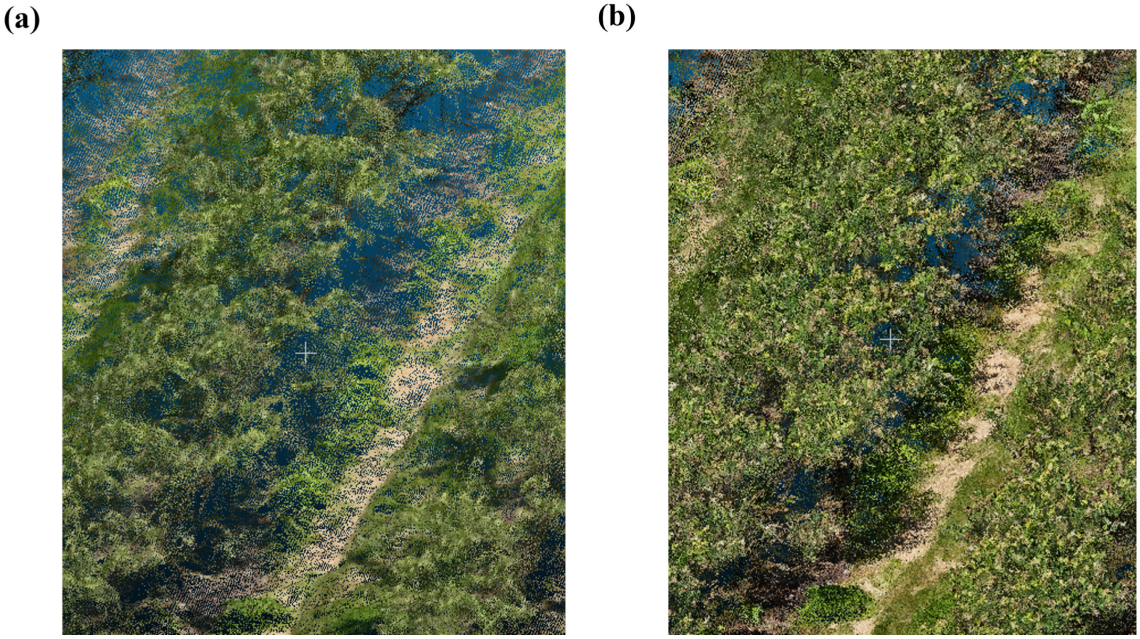

Figure 4.2. Zoomed section of experiment site point cloud (blueberry field, near-ground view): (a) conventional UAV data, showing limited under-canopy detail; (b) integrated UAV and nano-drone data, providing enhanced representation of ground-level structures.

To illustrate these differences, Figure 4.2 presents a zoomed section of the near-ground view. In the UAV-only data (Figure 4.2a), the lower canopy and ground structures are poorly represented, whereas in the integrated dataset (Figure 4.2b), the combination of UAV and nano-drone inputs captures significantly more detail at ground level.

To substantiate these qualitative observations, quantitative metrics were extracted and summarized in Table 4.1. The first metric, average point density, reflects the overall coverage of the reconstructed scene. The second metric, percentage of ground-level points, specifically evaluates how much structural detail was captured below the canopy. Here, ground level was defined as the relative elevation range of 0–0.5 m, a threshold chosen to encompass weeds and low-lying vegetation typically growing at or just above the soil surface, while excluding most of the crop canopy. Results show that the nano-drone dataset achieved the highest proportion of ground-level points (37.14%), while the UAV dataset captured fewer details below the canopy.

Table 4.1. Quantitative characteristics of different point clouds

	Average point density (pts/m2)	**Ground-level percentage (%) (range 0-0.5m)**
UAV	55074	34.23
Nano-drone	132589	37.14
Integrated	184909	34.60

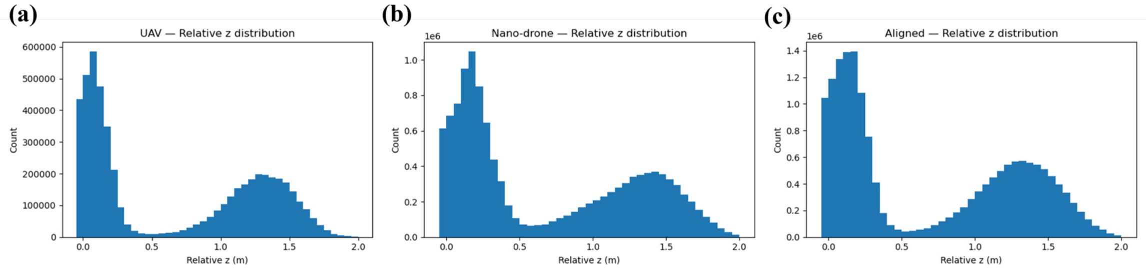

Figure 4.3. Vertical distribution histograms of point relative elevation of: (a) UAV data, (b) nano-drone data, (c) integrated dataset.

Figure 4.3 further supports this analysis with vertical distribution histograms of relative elevation. The UAV dataset (Figure 4.3a) shows a strong concentration at canopy height with minimal ground detail. The nano-drone dataset (Figure 4.3b) displays a broader distribution, particularly in the lower elevation range, while the integrated dataset (Figure 4.3c) presents a more continuous vertical profile, combining canopy and under-canopy information.

Overall, the results confirm that integrating UAV and nano-drone point clouds enhances both completeness and structural continuity of field reconstruction. By combining high canopy detail from UAV imagery with the enriched under-canopy detail from nano-drone imagery, the integrated dataset provides a superior foundation for weed detection in dense crop environments.

In the subsequent workflow, both the integrated point cloud and the UAV-only point cloud were processed through the same segmentation and classification steps. This parallel design allows for a direct comparison between the two datasets, with the goal of quantifying and demonstrating the benefits introduced by nano-drone–assisted mapping. By applying identical procedures to both inputs, the analysis highlights how the additional under-canopy detail captured by the nano-drone enhances weed identification performance in dense crop environments.

4.2. Crop Segmentation

The first step of segmentation involved applying the Progressive Morphological Filter (PMF) to separate the complete point cloud into ground and non-ground components. The ground point cloud included bare soil, weeds, and other low-lying vegetation, while the non-ground point cloud primarily represented the crop canopy. After segmentation, both components were color-coded to provide an intuitive visualization of the results, as shown in Figure 4.4 (a) and (d), where red indicates ground points and green corresponds to non-ground points.

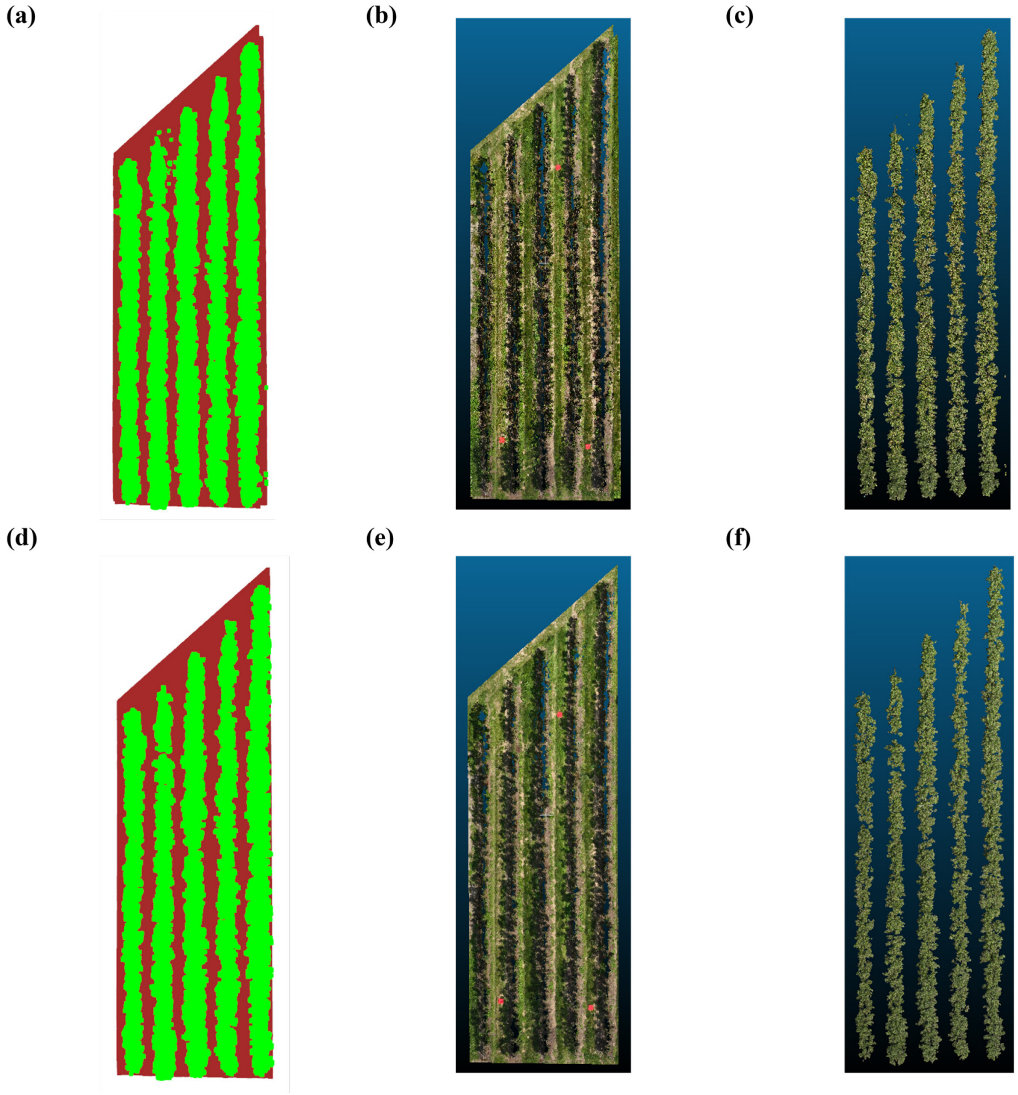

Figure 4.4. Visualized result of: (a) colorized binary segmentation map of integrated dataset, (b) ground point cloud of integrated dataset, (c) non-ground point cloud of integrated dataset, (d) colorized binary segmentation map of UAV-only dataset, (e) ground point cloud of UAV-only dataset, (f) non-ground point cloud of UAV-only dataset.

Both the integrated dataset and the UAV dataset produced similar crop segmentation outcomes. In both cases, the blueberry crop rows were clearly separated from the soil, and the overall structure of the plantation was well preserved. However, the integrated dataset exhibited slightly more noise points in the ground region. This effect arises from the imperfect fusion between the UAV and nano-drone point clouds, which introduced minor inconsistencies and layering artifacts in the ground surface. Despite this drawback, the integrated point cloud still provides richer detail for subsequent weed detection, especially in under-canopy regions where UAV-only data remain insufficient.

4.3. Individual Weed Segmentation

Following crop segmentation, soil–vegetation segmentation was performed on the ground point cloud using a weighted K-means scheme. The objective of this step was to remove all bare-soil points from the ground layer and retain only vegetation points (weeds and other low plants) as inputs to the subsequent individual-weed clustering.

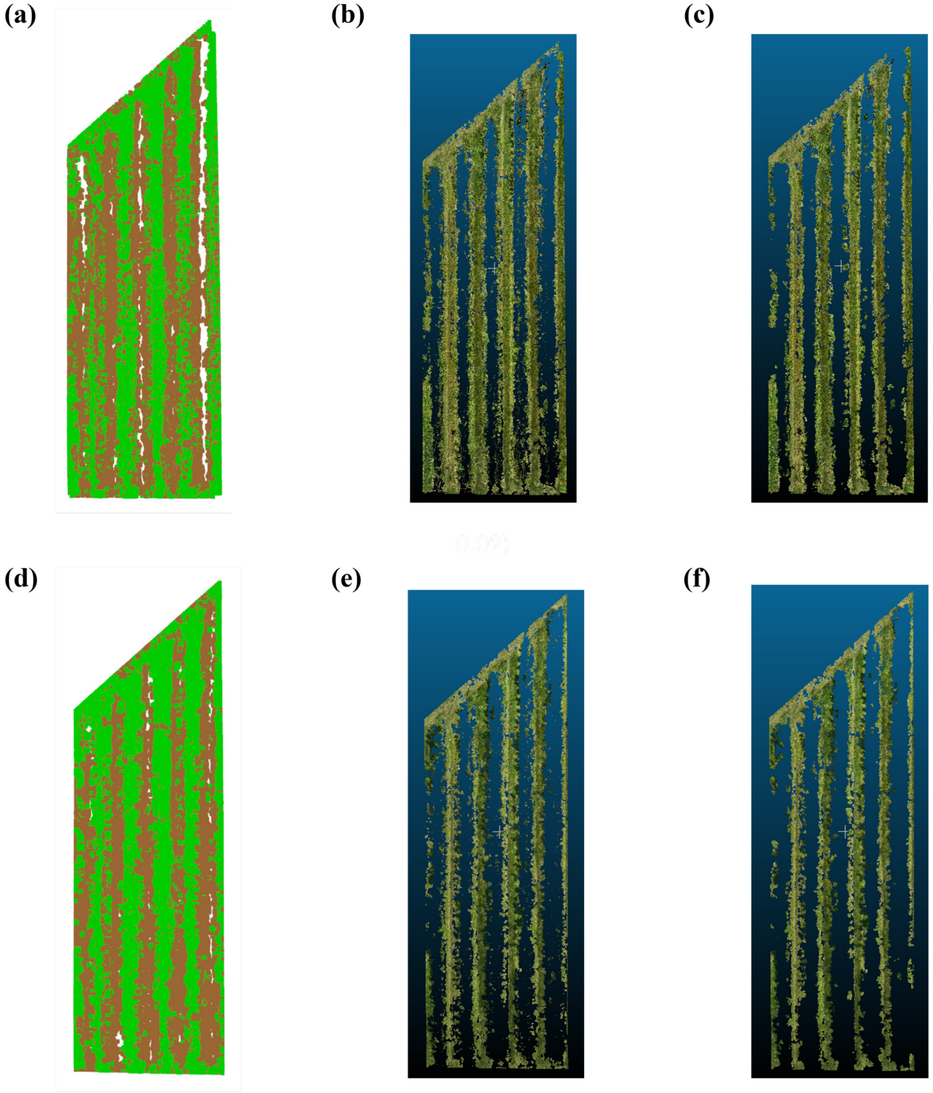

Figure 4.5. Visualized result of: (a) colorized binary segmentation map of integrated dataset, (b) segmented vegetation point cloud of integrated dataset, (c) denoised vegetation point cloud of integrated dataset, (d) colorized binary segmentation map of UAV-only dataset, (e) segmented vegetation point cloud of UAV-only dataset, (f) denoised vegetation point cloud of UAV-only dataset.

Figure 4.5 summarizes the intermediate results for the integrated dataset (top row, a–c) and the UAV-only dataset (bottom row, d–f). Panels (a) and (d) show the color-coded two-class maps (brown as soil, green as vegetation) produced by the weighted K-means segmentation. Panels (b) and (e) visualize the retained vegetation point clouds after removing soil. Panels (c) and (f) present the denoised vegetation clouds, where spurious points and small residual crop tips were suppressed before clustering.

Qualitatively, the soil–vegetation separation was successful for both datasets and aligns with the expected composition of the ground layer: bare soil beneath the canopy, grassy inter-row strips, and under-canopy weed patches preserved as vegetation. Comparing the two sources, the integrated dataset clearly exhibits richer under-canopy vegetation detail—more continuous patches and finer structures—reflecting the added visibility provided by nano-drone imagery.

The denoised vegetation point cloud obtained from the soil–vegetation segmentation served as the input for the next step of the individual weed segmentation pipeline. The process begins by applying DBSCAN to group vegetation points into clusters, followed by a K-means–based binary classification to separate weed clusters from non-weed vegetation. Finally, clusters identified as weeds are extracted for downstream weed map creation.

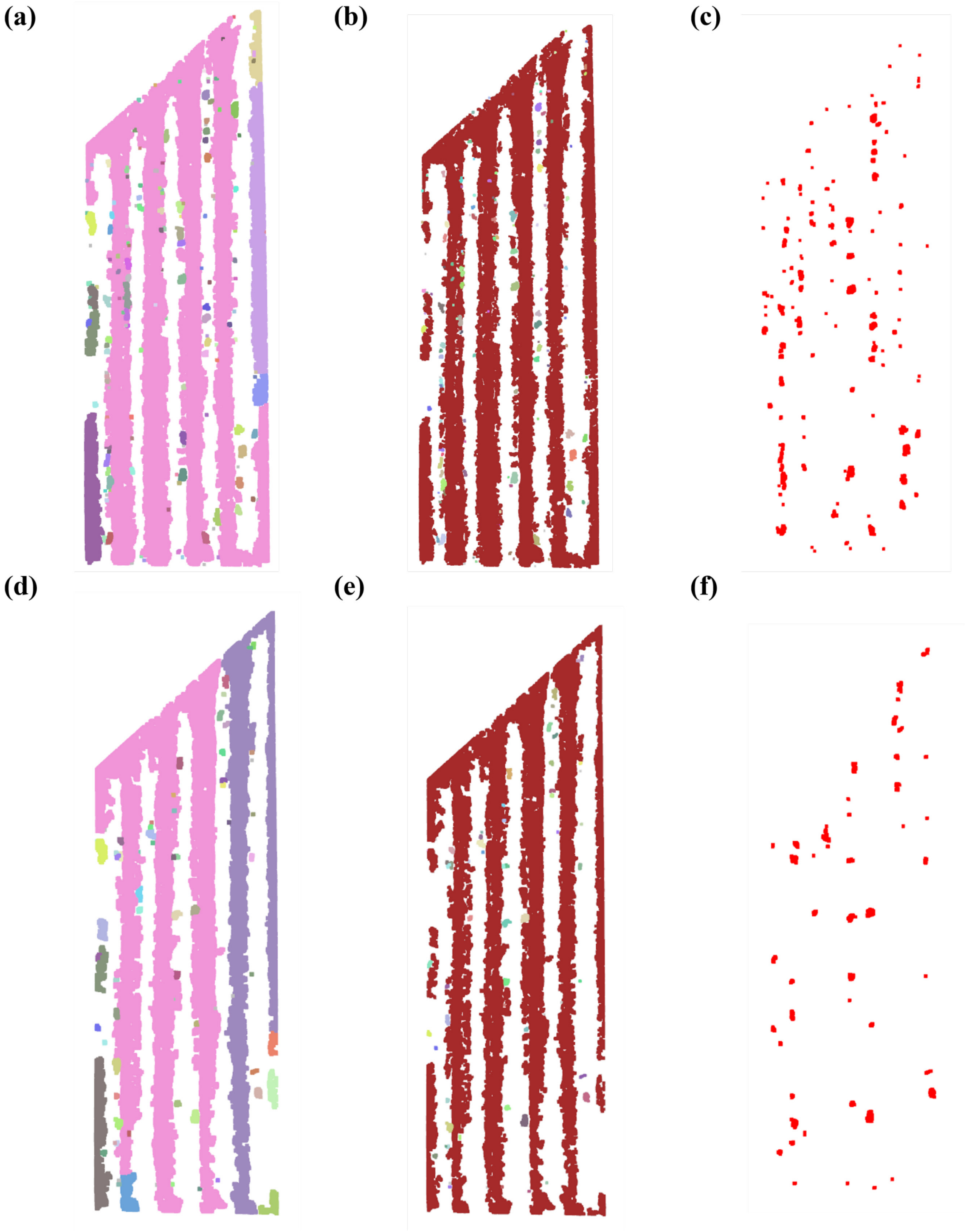

Figure 4.6. Visualized result of: (a) DBSCAN segmentation of integrated dataset, (b) colorized weed and non-weed of integrated dataset, (c) final weed clusters of integrated dataset, (d) DBSCAN segmentation of UAV-only dataset, (e) colorized weed and non-weed of UAV-only dataset, (f) final weed clusters of UAV-only dataset.

Figure 4.6 illustrates the workflow for the integrated dataset (a–c, top row) and the UAV-only dataset (d–f, bottom row). Panels (a) and (d) show the DBSCAN clustering outputs, where each cluster is visualized in a unique color. Panels (b) and (e) display the K-means classification results: red clusters represent non-weed vegetation, while multi-colored clusters indicate detected weed patches. Panels (c) and (f) show the final weed-only clusters extracted from the vegetation point cloud, which were later used for generating the weed distribution maps.

Qualitatively, both datasets capture inter-row and under-canopy vegetation, but the integrated dataset reveals a substantially richer weed distribution. Many under-canopy weeds that were invisible in the UAV-only point cloud were successfully detected in the integrated data. Quantitatively, the integrated dataset produced 125 distinct weed clusters, compared to only 52 clusters detected in the UAV-only dataset. This outcome confirms our core hypothesis: integrating nano-drone imagery with conventional UAV data significantly improves under-canopy weed detection by recovering structural details otherwise occluded from a top-down perspective.

4.4. Final Weed Map

The final weed distribution map is shown in Figure 4.7, where the green areas represent the 2D projection of crop points and the red areas denote the detected weed clusters. As illustrated, most of the weeds are located in the under-canopy region. This visualization clearly highlights the challenge of relying solely on conventional UAV imagery, as many of these weeds would be occluded from an overhead view.

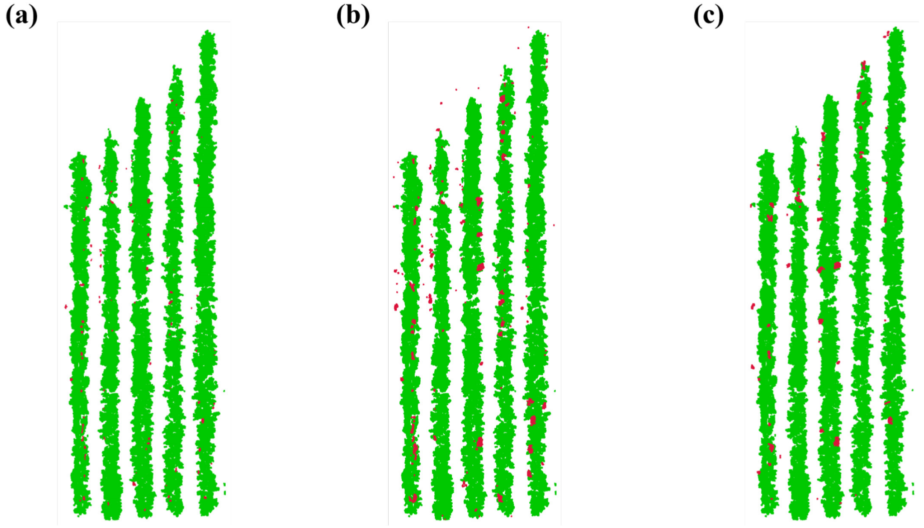

Figure 4.7. Weed distribution map of: (a) ground truth, (b) integrated dataset, (c) UAV-only dataset.

Table 4.2. Accuracy evaluation of weed detection

	Integrated dataset	UAV-only dataset
True positive	91	32
False positive	34	20
False negative	6	65
Precision	72.8%	61.5%
Recall	93.8%	33.0%
F1-score	82.1%	43.0%

Accuracy evaluation results are summarized in Table 4.2. The integrated dataset, which integrates UAV and nano-drone point clouds, achieved a precision of 72.8%, recall of 93.8%, and an overall F1-score of 82.1%. In contrast, the UAV-only dataset showed lower performance, with a precision of 61.5%, recall of 33.0%, and an F1-score of 43.0%. These results demonstrate that while the integrated dataset produced more false positives than UAV-only, it also detected substantially more true positives and missed far fewer weeds. The high recall value indicates that the integrated dataset provides a more complete representation of weed distribution, reducing the risk of undetected weeds in under-canopy areas.

Overall, the weed maps confirm the effectiveness of integrating nano-drone imagery with UAV data. The combined dataset not only enriches under-canopy detail but also improves detection accuracy, validating the hypothesis that multi-perspective point clouds enhance under-canopy weed identification.

Chapter 5. Conclusion and Future Work

5.1. Conclusion

This thesis developed and rigorously evaluated a nano-drone–assisted, unsupervised 3D point-cloud framework for under-canopy weed detection in perennial crop fields. The framework addressed the critical limitation of conventional UAV imagery, namely the occlusion of weeds beneath dense crop canopies, by integrating complementary aerial and near-ground perspectives.

The proposed workflow consisted of four main stages: (i) dual-perspective data acquisition, combining UAV canopy-level coverage with nano-drone near-ground observations; (ii) coarse alignment through DBSCAN-based GCP extraction and rigid transformation via triangle geometry; (iii) fine alignment using Iterative Closest Point (ICP); and (iv) multi-level segmentation, including PMF-based ground/non-ground separation, weighted K-means soil–vegetation classification, and DBSCAN–K-means clustering for individual weed identification. The final output was a 2D weed distribution map that directly supports site-specific field management.

Field experiments conducted in a densely planted blueberry farm confirmed that the integrated multi-perspective dataset significantly outperformed UAV-only data. The integrated dataset identified 125 weed clusters, more than double the UAV-only result of 52 clusters. Accuracy metrics further reinforced this improvement, with precision of 72.8%, recall of 93.8%, and F1-score of 82.1% compared with 61.5%, 33.0%, and 43.0% for UAV-only data. Vertical distribution histograms highlighted the structural complementarity of the two datasets, validating the hypothesis that multi-perspective integration enhances weed detection in occluded environments.

Key contributions of this thesis can be summarized as follows:

- Novel integration of UAV and nano-drone imagery: Demonstrated the feasibility and effectiveness of combining top-down and under-canopy perspectives for agricultural point-cloud reconstruction.

- Robust point-cloud alignment pipeline: Developed a practical two-step alignment strategy (GCP-based transformation + ICP) that ensured structural consistency between heterogeneous datasets.

- Novel unsupervised weed detection framework: Implemented a multi-stage segmentation process capable of separating soil, crop, and weed clusters without reliance on labeled data.

- Validated improvement in detection accuracy: Provided quantitative evidence that multi-perspective integration nearly doubled the number of identified weed clusters and significantly improved precision, recall, and F1-score compared with UAV-only workflows.

Despite these achievements, several limitations remain, including residual misalignment artifacts in the fused ground layer, parameter sensitivity for density-based clustering, and false positives arising in heterogeneous ground conditions. These challenges highlight the need for improved data acquisition stability, registration refinement, and post-processing strategies in future work.

5.2. Future Work

Building upon the contributions of this research, several directions for future investigation are identified:

(1) Enhancement of georeferencing accuracy and data acquisition stability.

Improving the positional accuracy of UAV platforms, particularly nano-drones, is critical for reducing residual misalignment and ensuring reliable reconstructions. Future work should explore the integration of RTK/PPK GNSS systems, denser GCP deployment, and synchronization of camera and IMU/GNSS measurements. Additionally, improvements in nano-drone stability—through enhanced waypoint navigation, visual-inertial odometry, or terrain-following flight modes—may further increase the reliability of near-ground acquisitions.

(2) Development of large-scale weed density mapping tools.

Beyond cluster-level detection, the framework can be extended to generate field-scale weed density maps, enabling farmers to identify infestation hotspots and assess spatial variability. Grid-based or row-based density representations, combined with temporal monitoring, would provide valuable decision support for adaptive weed management strategies.

(3) Integration with autonomous weeding systems.

A longer-term objective is to establish a practical, closed-loop weed-management workflow in which UAV-acquired imagery and the proposed under-canopy detection framework directly support robotic weeding platforms. In such a system, farmers would follow a simple operational sequence:

1. deploy a conventional UAV and a nano-drone to capture canopy-level and under-canopy images,
2. upload the raw images into a user-friendly software package,
3. automatically generate a fused 3D point cloud and corresponding weed distribution map, and
4. send the georeferenced weed map to an autonomous weeding robot for targeted intervention.

Compared with using a robot alone—which must physically explore the full field to perform real-time weed detection—integrating pre-generated weed maps dramatically improves operational efficiency. The robot no longer needs to survey the entire field visually; it only navigates to predetermined weed locations and performs localized removal. This reduces energy consumption, decreases field traversal time, minimizes mechanical wear, and increases the robot's effective daily treatment capacity.

To support real-world adoption, the full framework will be encapsulated into a streamlined software platform. The tool will accept raw UAV and nano-drone imagery as inputs, automatically execute alignment, segmentation, and weed-clustering algorithms in the background, and output a standardized georeferenced weed map. Once generated, the map can be transmitted directly to a weeding robot through standard communication protocols, enabling rapid "scan → analyze → act" operation. Such a pipeline lowers the technical barrier for farmers, enhances decision-making efficiency, and moves toward a fully automated, precise, and sustainable weed-management system.

References

[1] R. L. Zimdahl, *Fundamentals of Weed Science*. Elsevier, 2007, pp. 13–28.

[2] M. K. Upadhyaya and R. E. Blackshaw, *Non-chemical weed management : principles, concepts and technology*. Wallingford, Uk ; Cambridge, Ma: Cabi, 2007.

[3] Santosh Kumar Paul, S. Mazumder, and R. Naidu, "Herbicidal weed management practices: History and future prospects of nanotechnology in an eco-friendly crop production system," *Heliyon*, vol. 10, no. 5, pp. e26527–e26527, Mar. 2024, doi: https://doi.org/10.1016/j.heliyon.2024.e26527.

[4] Z. P. ZHANG, "Development of chemical weed control and integrated weed management in China," *Weed Biology and Management*, vol. 3, no. 4, pp. 197–203, Dec. 2003, doi: https://doi.org/10.1046/j.1444-6162.2003.00105.x.

[5] J. H. 1 Westwood *et al.*, "Weed Management in 2050: Perspectives on the Future of Weed Science," *ProQuest*, pp. 275–285, May 2018, doi: https://doi.org/10.1017/wsc.2017.78.

[6] M. D. Owen and I. A. Zelaya, "Herbicide-resistant crops and weed resistance to herbicides," *Pest Management Science*, vol. 61, no. 3, pp. 301–311, 2005, doi: https://doi.org/10.1002/ps.1015.

[7] M. A. Peterson, A. Collavo, R. Ovejero, V. Shivrain, and M. J. Walsh, "The challenge of herbicide resistance around the world: a current summary," *Pest Management Science*, vol. 74, no. 10, pp. 2246–2259, Feb. 2018, doi: https://doi.org/10.1002/ps.4821.

[8] M. Graymore, F. Stagnitti, and G. Allinson, "Impacts of atrazine in aquatic ecosystems," *Environment International*, vol. 26, no. 7–8, pp. 483–495, Jun. 2001, doi: https://doi.org/10.1016/s0160-4120(01)00031-9.

[9] D. Seghers *et al.*, "Effect of long-term herbicide applications on the bacterial community structure and function in an agricultural soil," *FEMS microbiology ecology/FEMS microbiology, ecology*, vol. 46, no. 2, pp. 139–146, Nov. 2003, doi: https://doi.org/10.1016/s0168-6496(03)00205-8.

[10] A. H. C. Van Bruggen *et al.*, "Environmental and health effects of the herbicide glyphosate," *Science of The Total Environment*, vol. 616–617, pp. 255–268, Mar. 2018, doi: https://doi.org/10.1016/j.scitotenv.2017.10.309.

[11] A. Parven, I. Md Meftaul, K. Venkateswarlu, and M. Megharaj, "Herbicides in modern sustainable agriculture: environmental fate, ecological implications, and human health concerns," *International journal of environmental science and technology*, vol. 22, no. 2, Jun. 2024, doi: https://doi.org/10.1007/s13762-024-05818-y.

[12] A. Monteiro and S. Santos, "Sustainable Approach to Weed Management: The Role of Precision Weed Management," *Agronomy*, vol. 12, no. 1, p. 118, Jan. 2022, doi: https://doi.org/10.3390/agronomy12010118.

[13] L. R. Newcome, *Unmanned aviation : a brief history of unmanned aerial vehicles*. Reston, Va.: American Institute Of Aeronautics And Astronautics, Cop, 2004, pp. 1–3.

[14] S. G. K. Kumar and A. Gudipalli, “A comprehensive review on payloads of unmanned aerial vehicle,” *The Egyptian Journal of Remote Sensing and Space Science*, vol. 27, no. 4, pp. 637–644, Aug. 2024, doi: https://doi.org/10.1016/j.ejrs.2024.08.001.

[15] H. Ren, Y. Zhao, W. Xiao, and Z. Hu, “A review of UAV monitoring in mining areas: current status and future perspectives,” *International Journal of Coal Science & Technology*, vol. 6, no. 3, pp. 320–333, Aug. 2019, doi: https://doi.org/10.1007/s40789-019-00264-5.

[16] C. Torresan *et al.*, “Forestry applications of UAVs in Europe: a review,” *International Journal of Remote Sensing*, vol. 38, no. 8–10, pp. 2427–2447, Nov. 2016, doi: https://doi.org/10.1080/01431161.2016.1252477.

[17] D. C. Tsouros, S. Bibi, and P. G. Sarigiannidis, “A Review on UAV-Based Applications for Precision Agriculture,” *Information*, vol. 10, no. 11, p. 349, Nov. 2019, doi: https://doi.org/10.3390/info10110349.

[18] J. Kim, S. Kim, C. Ju, and H. I. Son, “Unmanned Aerial Vehicles in Agriculture: A Review of Perspective of Platform, Control, and Applications,” *IEEE Access*, vol. 7, pp. 105100–105115, 2019, doi: https://doi.org/10.1109/ACCESS.2019.2932119.

[19] F. Remondino, L. Barazzetti, F. Nex, M. Scaioni, and D. Sarazzi, “UAV PHOTOGRAMMETRY FOR MAPPING AND 3D MODELING – CURRENT STATUS AND FUTURE PERSPECTIVES,” *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVIII-1/C22, pp. 25–31, Sep. 2012, doi: https://doi.org/10.5194/isprsarchives-xxxviii-1-c22-25-2011.

[20] F. Nex and F. Remondino, “UAV for 3D mapping applications: a review,” *Applied Geomatics*, vol. 6, no. 1, pp. 1–15, Nov. 2013, doi: https://doi.org/10.1007/s12518-013-0120-x.

[21] M. J. Westoby, J. Brasington, N. F. Glasser, M. J. Hambrey, and J. M. Reynolds, “‘Structure-from-Motion’ photogrammetry: A low-cost, effective tool for geoscience applications,” *Geomorphology*, vol. 179, pp. 300–314, Dec. 2012.

[22] J. Ni, L. Yao, J. Zhang, W. Cao, Y. Zhu, and X. Tai, “Development of an Unmanned Aerial Vehicle-Borne Crop-Growth Monitoring System,” *Sensors*, vol. 17, no. 3, p. 502, Mar. 2017, doi: https://doi.org/10.3390/s17030502.

[23] Z. Fu *et al.*, “Wheat Growth Monitoring and Yield Estimation based on Multi-Rotor Unmanned Aerial Vehicle,” *Remote Sensing*, vol. 12, no. 3, p. 508, Feb. 2020, doi: https://doi.org/10.3390/rs12030508.

[24] N. Yu, L. Li, N. Schmitz, L. F. Tian, J. A. Greenberg, and B. W. Diers, “Development of methods to improve soybean yield estimation and predict plant maturity with an unmanned aerial vehicle based platform,” *Remote Sensing of Environment*, vol. 187, pp. 91–101, Dec. 2016, doi: https://doi.org/10.1016/j.rse.2016.10.005.

[25] Y. Huang, K. N. Reddy, R. S. Fletcher, and D. Pennington, “UAV Low-Altitude Remote Sensing for Precision Weed Management,” *Weed Technology*, vol. 32, no. 1, pp. 2–6, Nov. 2017, doi: https://doi.org/10.1017/wet.2017.89.

[26] T. Xie *et al.*, “Crop height estimation based on UAV images: Methods, errors, and strategies,” *Computers and Electronics in Agriculture*, vol. 185, p. 106155, Jun. 2021, doi: https://doi.org/10.1016/j.compag.2021.106155.

[27] Y. Qi *et al.*, “Canopy Volume Extraction of Citrus reticulate Blanco cv. Shatangju Trees Using UAV Image-Based Point Cloud Deep Learning,” *Remote Sensing*, vol. 13, no. 17, pp. 3437–3437, Aug. 2021, doi: https://doi.org/10.3390/rs13173437.

[28] Z. Wu, Y. Chen, B. Zhao, X. Kang, and Y. Ding, “Review of Weed Detection Methods Based on Computer Vision,” *Sensors*, vol. 21, no. 11, p. 3647, May 2021, doi: https://doi.org/10.3390/s21113647.

[29] G. R. Coleman *et al.*, “Weed detection to weed recognition: reviewing 50 years of research to identify constraints and opportunities for large-scale cropping systems,” *Weed Technology*, pp. 1–50, Nov. 2022, doi: https://doi.org/10.1017/wet.2022.84.

[30] S. T, S. T, G.S., Sukanya Gowthami, S. S, and R. Kumaraswamy, “Performance comparison of weed detection algorithms,” 2019, pp. 0843–0847. doi: https://doi.org/10.1109/ICCSP.2019.8698094.

[31] W. Zhang, Z. Miao, N. Li, C. He, and T. Sun, “Review of Current Robotic Approaches for Precision Weed Management,” *Current Robotics Reports*, vol. 3, no. 3, pp. 139–151, Jul. 2022, doi: https://doi.org/10.1007/s43154-022-00086-5.

[32] N. Shahbazi *et al.*, “Assessing the Capability and Potential of LiDAR for Weed Detection,” *Sensors*, vol. 21, no. 7, p. 2328, Mar. 2021, doi: https://doi.org/10.3390/s21072328.

[33] U. Weiss and P. Biber, “Plant detection and mapping for agricultural robots using a 3D LIDAR sensor,” *Robotics and Autonomous Systems*, vol. 59, no. 5, pp. 265–273, May 2011, doi: https://doi.org/10.1016/j.robot.2011.02.011.

[34] T. Canada, “Canadian Aviation Regulations (SOR/96-433),” *AARBH 14882767*, Jan. 25, 2021. https://tc.canada.ca/en/corporate-services/acts-regulations/list-regulations/canadian-aviation-regulations-sor-96-433

[35] L. Petricca, P. Ohlckers, and C. Grinde, “Micro- and Nano-Air Vehicles: State of the Art,” *International Journal of Aerospace Engineering*, vol. 2011, pp. 1–17, 2011, doi: https://doi.org/10.1155/2011/214549.

[36] Kosta Gligorević, M. Dražić, Miloš Pajić, M. Šunjevarić, Biljana Bošković, and Mićo Oljača, “Overview of the possibility application of some nano drone technologies in modern agriculture,” *Poljoprivredna tehnika*, vol. 49, no. 1, pp. 75–96, Jan. 2024, doi: https://doi.org/10.5937/poljteh2401075g.

[37] R. N. Abutalipov, Y. V. Bolgov, and H. M. Senov, “Flowering plants pollination robotic system for greenhouses by means of nano copter (drone aircraft),” *2016 IEEE Conference on Quality Management, Transport and Information Security, Information Technologies (IT&MQ&IS)*, Oct. 2016, doi: https://doi.org/10.1109/itmqis.2016.7751907.

[38] J. Gago *et al.*, “Nano and Micro Unmanned Aerial Vehicles (UAVs): A New Grand Challenge for Precision Agriculture?,” *Current Protocols in Plant Biology*, vol. 5, no. 1, Feb. 2020, doi: https://doi.org/10.1002/cppb.20103.

[39] B. Li *et al.*, “The estimation of crop emergence in potatoes by UAV RGB imagery,” *Plant Methods*, vol. 15, no. 1, p. 15, 2019, doi: https://doi.org/10.1186/s1300701903997.

[40] J. Jung, M. Maeda, A. Chang, J. Landivar, J. Yeom, and J. McGinty, “Unmanned aerial system assisted framework for the selection of high yielding cotton genotypes,” *Computers and Electronics in Agriculture*, vol. 152, pp. 74–81, Sep. 2018, doi: https://doi.org/10.1016/j.compag.2018.06.051.

[41] J. Senthilnath, A. Dokania, M. Kandukuri, R. K.N., G. Anand, and S. N. Omkar, “Detection of tomatoes using spectral-spatial methods in remotely sensed RGB images captured by UAV,” *Biosystems Engineering*, vol. 146, pp. 16–32, Jun. 2016, doi: https://doi.org/10.1016/j.biosystemseng.2015.12.003.

[42] S. Guan *et al.*, “Assessing Correlation of High-Resolution NDVI with Fertilizer Application Level and Yield of Rice and Wheat Crops Using Small UAVs,” *Remote Sensing*, vol. 11, no. 2, p. 112, Jan. 2019, doi: https://doi.org/10.3390/rs11020112.

[43] K. Johansen *et al.*, “Mapping the condition of macadamia tree crops using multispectral UAV and WorldView3 imagery,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 165, pp. 28–40, 2020, doi: https://doi.org/10.1016/j.isprsjprs.2020.04.017.

[44] P. J. ZarcoTejada, M. L. GuillénCliment, R. HernándezClemente, A. Catalina, M. R. González, and P. Martín, “Estimating leaf carotenoid content in vineyards using high resolution hyperspectral imagery acquired from an unmanned aerial vehicle (UAV),” *Agricultural and Forest Meteorology*, vol. 171172, pp. 281–294, 2013, doi: https://doi.org/10.1016/j.agrformet.2012.12.013.

[45] A. Lucieer, Z. Malenovský, T. Veness, and L. Wallace, “HyperUAS-Imaging Spectroscopy from a Multirotor Unmanned Aircraft System,” *Journal of Field Robotics*, vol. 31, no. 4, pp. 571–590, Mar. 2014, doi: https://doi.org/10.1002/rob.21508.

[46] S. Park, D. Ryu, S. Fuentes, H. Chung, E. Hernández-Montes, and M. O’Connell, “Adaptive Estimation of Crop Water Stress in Nectarine and Peach Orchards Using High-Resolution Imagery from an Unmanned Aerial Vehicle (UAV),” *Remote Sensing*, vol. 9, no. 8, p. 828, Aug. 2017, doi: https://doi.org/10.3390/rs9080828.

[47] M. Christiansen, M. Laursen, R. Jørgensen, S. Skovsen, and R. Gislum, “Designing and Testing a UAV Mapping System for Agricultural Field Surveying,” *Sensors*, vol. 17, no. 12, p. 2703, Nov. 2017, doi: https://doi.org/10.3390/s17122703.

[48] W. Sangjan, R. J. McGee, and S. Sankaran, “Optimization of UAV-Based Imaging and Image Processing Orthomosaic and Point Cloud Approaches for Estimating Biomass in a Forage Crop,” *Remote Sensing*, vol. 14, no. 10, p. 2396, May 2022, doi: https://doi.org/10.3390/rs14102396.

[49] A. Wang, W. Zhang, and X. Wei, “A review on weed detection using ground-based machine vision and image processing techniques,” *Computers and Electronics in Agriculture*, vol. 158, pp. 226–240, Mar. 2019, doi: https://doi.org/10.1016/j.compag.2019.02.005.

[50] K. Osorio, A. Puerto, C. Pedraza, D. Jamaica, and L. Rodríguez, “A Deep Learning Approach for Weed Detection in Lettuce Crops Using Multispectral Images,” *AgriEngineering*, vol. 2, no. 3, pp. 471–488, Aug. 2020, doi: https://doi.org/10.3390/agriengineering2030032.

[51] O. Barrero and S. A. Perdomo, “RGB and multispectral UAV image fusion for Gramineae weed detection in rice fields,” *Precision Agriculture*, vol. 19, no. 5, pp. 809–822, Jan. 2018, doi: https://doi.org/10.1007/s11119-017-9558-x.

[52] X.-E. Pantazi, D. Moshou, and C. Bravo, “Active learning system for weed species recognition based on hyperspectral sensing,” *Biosystems Engineering*, vol. 146, pp. 193–202, Jun. 2016, doi: https://doi.org/10.1016/j.biosystemseng.2016.01.014.

[53] S. Murawwat, A. Qureshi, S. Ahmad, and Y. Shahid, “Weed Detection Using SVMs,” *Engineering, Technology & Applied Science Research*, vol. 8, no. 1, pp. 2412–2416, Feb. 2018, doi: https://doi.org/10.48084/etasr.1647.

[54] A. de Castro, J. Torres-Sánchez, J. Peña, F. Jiménez-Brenes, O. Csillik, and F. López-Granados, “An Automatic Random Forest-OBIA Algorithm for Early Weed Mapping between and within Crop Rows Using UAV Imagery,” *Remote Sensing*, vol. 10, no. 3, p. 285, Feb. 2018, doi: https://doi.org/10.3390/rs10020285.

[55] A. L. Monteiro *et al.*, “A new alternative to determine weed control in agricultural systems based on artificial neural networks (ANNs),” *Field Crops Research*, vol. 263, p. 108075, Apr. 2021, doi: https://doi.org/10.1016/j.fcr.2021.108075.

[56] M. Anul Haq, “CNN Based Automated Weed Detection System Using UAV Imagery,” *Computer Systems Science and Engineering*, vol. 42, no. 2, pp. 837–849, 2022, doi: https://doi.org/10.32604/csse.2022.023016.

[57] J. Chen *et al.*, “Weed detection in sesame fields using a YOLO model with an enhanced attention mechanism and feature fusion,” *Computers and Electronics in Agriculture*, vol. 202, pp. 107412–107412, Nov. 2022, doi: https://doi.org/10.1016/j.compag.2022.107412.

[58] X. Zhang *et al.*, “Weed Identification in Soybean Seedling Stage Based on Optimized Faster R-CNN Algorithm,” *Agriculture*, vol. 13, no. 1, p. 175, Jan. 2023, doi: https://doi.org/10.3390/agriculture13010175.

[59] A. dos Santos Ferreira, D. M. Freitas, G. G. da Silva, H. Pistori, and M. T. Folhes, “Unsupervised deep learning and semi-automatic data labeling in weed discrimination,” *Computers and Electronics in Agriculture*, vol. 165, p. 104963, Oct. 2019, doi: https://doi.org/10.1016/j.compag.2019.104963.

[60] D. Andújar, J. Dorado, C. Fernández-Quintanilla, and A. Ribeiro, “An Approach to the Use of Depth Cameras for Weed Volume Estimation,” *Sensors*, vol. 16, no. 7, p. 972, Jun. 2016, doi: https://doi.org/10.3390/s16070972.

[61] A. Piron, F. van der Heijden, and M. F. Destain, “Weed detection in 3D images,” *Precision Agriculture*, vol. 12, no. 5, pp. 607–622, Nov. 2010, doi: https://doi.org/10.1007/s11119-010-9205-2.

[62] A. M. Dobbs *et al.*, “New directions in weed management and research using 3D imaging,” *Weed Science*, vol. 70, no. 6, p. 641647, 2022, doi: https://doi.org/10.1017/wsc.2022.56.

[63] T. Tao, S. Wu, L. Li, J. Li, S. Bao, and X. Wei, “Design and experiments of weeding teleoperated robot spectral sensor for winter rape and weed identification,” *Advances in Mechanical Engineering*, vol. 10, no. 5, p. 168781401877674-168781401877674, May 2018, doi: https://doi.org/10.1177/1687814018776741.

[64] S. Sabzi, Y. Abbaspour-Gilandeh, and J. I. Arribas, “An automatic visible-range video weed detection, segmentation and classification prototype in potato field,” *Heliyon*, vol. 6, no. 5, p. e03685, May 2020, doi: https://doi.org/10.1016/j.heliyon.2020.e03685.

[65] W. Saif, *Photogrammetry: A Brief Historical Overview*. 2022. doi: https://doi.org/10.13140/RG.2.2.27518.87369.

[66] P. J. Schneider and D. H. Eberly, *Geometric tools for computer graphics*. San Francisco: Morgan Kaufmann, 2003, pp. 481–662.

[67] C. Cao, M. Preda, and T. Zaharia, “3D Point Cloud Compression,” *Proceedings of the 24th International Conference on 3D Web Technology*, Jul. 2019, doi: https://doi.org/10.1145/3329714.3338130.

[68] A. Wehr and U. Lohr, “Airborne laser scanning—an introduction and overview,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 54, no. 2–3, pp. 68–82, Jul. 1999, doi: https://doi.org/10.1016/s0924-2716(99)00011-8.

[69] Q. Wang and M.-K. Kim, “Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018,” *Advanced Engineering Informatics*, vol. 39, pp. 306–319, Jan. 2019, doi: https://doi.org/10.1016/j.aei.2019.02.007.

[70] J. Trochta, M. Krůček, T. Vrška, and K. Král, “3D Forest: An application for descriptions of three-dimensional forest structures using terrestrial LiDAR,” *PLOS ONE*, vol. 12, no. 5, p. e0176871, May 2017, doi: https://doi.org/10.1371/journal.pone.0176871.

[71] X. Liu, W. Zhu, X. Lian, and X. Xu, “Monitoring Mining Surface Subsidence with Multi-Temporal Three-Dimensional Unmanned Aerial Vehicle Point Cloud,” *Remote Sensing*, vol. 15, no. 2, p. 374, Jan. 2023, doi: https://doi.org/10.3390/rs15020374.

[72] L. Comba, A. Biglia, D. Ricauda Aimonino, and P. Gay, “Unsupervised detection of vineyards by 3D point-cloud UAV photogrammetry for precision agriculture,” *Computers and Electronics in Agriculture*, vol. 155, pp. 84–95, Dec. 2018, doi: https://doi.org/10.1016/j.compag.2018.10.005.

[73] E. Hyyppä *et al.*, “Under-canopy UAV laser scanning for accurate forest field measurements,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 164, pp. 41–60, Jun. 2020, doi: https://doi.org/10.1016/j.isprsjprs.2020.03.021.

[74] X.-F. Han, J. S. Jin, M.-J. Wang, W. Jiang, L. Gao, and L. Xiao, “A review of algorithms for filtering the 3D point cloud,” *Signal Processing: Image Communication*, vol. 57, pp. 103–112, Sep. 2017, doi: https://doi.org/10.1016/j.image.2017.05.009.

[75] S. Gumhold, X. Wang, and R. Macleod, “Feature Extraction from Point Clouds,” *Proceedings of 10th international meshing roundtable*, vol. 2001, Nov. 2001.

[76] Z. Huang, Y. Wen, Z. Wang, J. Ren, and K. Jia, “Surface Reconstruction From Point Clouds: A Survey and a Benchmark,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 9727–9748, doi: https://doi.org/10.1109/TPAMI.2024.3429209.

[77] A. Nurunnabi, D. Belton, and G. West, “DIAGNOSTIC-ROBUST STATISTICAL ANALYSIS FOR LOCAL SURFACE FITTING IN 3D POINT CLOUD DATA,” *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. I–3, pp. 269–274, Jul. 2012, doi: https://doi.org/10.5194/isprsannals-i-3-269-2012.

[78] R. B. Rusu and S. Cousins, “3D is here: Point Cloud Library (PCL),” in *2011 IEEE International Conference on Robotics and Automation*, pp. 1–4. doi: https://doi.org/10.1109/ICRA.2011.5980567.

[79] Q.-Y. Zhou, J. Park, and V. Koltun, “Open3D: A Modern Library for 3D Data Processing,” *arXiv.org*, Jan. 29, 2018. https://arxiv.org/abs/1801.09847

[80] X. Huang, G. Mei, and J. Zhang, “Crosssource point cloud registration: Challenges, progress and prospects,” *Neurocomputing*, vol. 548, p. 126383, 2023, doi: https://doi.org/10.1016/j.neucom.2023.126383.

[81] P. J. Besl and N. D. McKay, “A method for registration of 3-D shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, Feb. 1992, doi: https://doi.org/10.1109/34.121791.

[82] J. C. Gower and G. B. Dijksterhuis, *Procrustes Problems*. OUP Oxford, 2004.

[83] R. Toldo, A. Beinat, and F. Crosilla, “Global registration of multiple point clouds embedding the Generalized Procrustes Analysis into an ICP framework,” *International Symposium on 3D Data Processing, Visualization and Transmission*, Jan. 2010.

[84] J. Yang, Z. Cao, and Q. Zhang, “A fast and robust local descriptor for 3D point cloud registration,” *Information Sciences*, vol. 346347, pp. 163–179, 2016, doi: https://doi.org/10.1016/j.ins.2016.01.095.

[85] X. Huang, J. Zhang, Q. Wu, L. Fan, and C. Yuan, “A CoarsetoFine Algorithm for Registration in 3D StreetView CrossSource Point Clouds,” in *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–6. doi: https://doi.org/10.1109/DICTA.2016.7796986.

[86] Y. Aoki, H. Goforth, Rangaprasad Arun Srivatsan, and S. Lucey, “PointNetLK: Robust & Efficient Point Cloud Registration using PointNet,” *arXiv (Cornell University)*, Jan. 2019, doi: https://doi.org/10.48550/arxiv.1903.05711.

[87] Y. Wang and J. Solomon, “Deep Closest Point: Learning Representations for Point Cloud Registration,” *International Conference on Computer Vision*, May 2019, doi: https://doi.org/10.1109/iccv.2019.00362.

[88] A. Kurobe, Y. Sekikawa, K. Ishikawa, and H. Saito, “CorsNet: 3D Point Cloud Registration by Deep Neural Network,” *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3960–3966, doi: https://doi.org/10.1109/LRA.2020.2970946.

[89] E. Grilli, F. Menna, and F. Remondino, “A REVIEW OF POINT CLOUDS SEGMENTATION AND CLASSIFICATION ALGORITHMS,” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information*

Sciences, vol. XLII-2/W3, pp. 339–344, 2017, doi: https://doi.org/10.5194/isprs-archives-XLII-2-W3-339-2017.

[90] J. Zhang, X. Zhao, Z. Chen, and Z. Lu, "A Review of Deep LearningBased Semantic Segmentation for Point Cloud," *IEEE Access*, vol. 7, pp. 179118–179133, 2019, doi: https://doi.org/10.1109/ACCESS.2019.2958671.

[91] J. Zhou, X. Fu, S. Zhou, J. Zhou, H. Ye, and H. T. Nguyen, "Automated segmentation of soybean plants from 3D point cloud using machine learning," *Computers and Electronics in Agriculture*, vol. 162, pp. 143–153, Jul. 2019, doi: https://doi.org/10.1016/j.compag.2019.04.014.

[92] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, vol. 1, pp. 281–297, 1967.

[93] J. Yadav and M. Sharma, "A review of k-mean algorithm," *Int. J. Eng. Trends Technol*, vol. 4, Art. no. 7, 2013.

[94] K. P. Sinaga and M.-S. Yang, "Unsupervised K-Means Clustering Algorithm," *IEEE Access*, vol. 8, no. 1, pp. 80716–80727, 2020, doi: https://doi.org/10.1109/access.2020.2988796.

[95] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Knowledge Discovery and Data Mining*, 1996, pp. 226–231.

[96] H. AverbuchElor, N. Bar, and D. CohenOr, "Border-Peeling Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 7, pp. 1791–1797, doi: https://doi.org/10.1109/TPAMI.2019.2924953.

[97] N. Brodu and D. Lague, "3D terrestrial lidar data classification of complex natural scenes using a multiscale dimensionality criterion: Applications in geomorphology," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 68, pp. 121–134, 2012, doi: https://doi.org/10.1016/j.isprsjprs.2012.01.006.

[98] H. Woo, E. Kang, S. Wang, and K. H. Lee, "A new segmentation method for point cloud data," *International Journal of Machine Tools and Manufacture*, vol. 42, no. 2, pp. 167–178, Jan. 2002, doi: https://doi.org/10.1016/s0890-6955(01)00120-1.

[99] A. TERRAIN, "Digital terrain models: an overview," *Photogrammetric Engineering and Remote Sensing*, vol. 44, Art. no. 12, 1978.

[100] A. Iqbal, M. Shahjahan Mondal, W. Veerbeek, M. Shah, and H. Hakvoort, "Effectiveness of UAV-based DTM and satellite-based DEMs for local-level flood modeling in Jamuna floodplain," *Journal of flood risk management*, vol. 16, no. 4, Jul. 2023, doi: https://doi.org/10.1111/jfr3.12937.

[101] Khamarrul Azahari Razak, M. Santangelo, C.J. van Westen, N. Wanders, and Steven de Jong, "Generating an optimal DTM from airborne laser scanning data for landslide mapping in a tropical forest environment," *Geomorphology*, vol. 190, pp. 112–125, May 2013, doi: https://doi.org/10.1016/j.geomorph.2013.02.021.

[102] S. W. Kienzle, "Using DTMs and GIS to define input variables for hydrological and geomorphological analysis," *IAHS Publications-Series of Proceedings and Reports-Intern Assoc Hydrological Sciences*, vol. 235, pp. 183–190, 1996.

[103] S. Jayathunga, T. Owari, and S. Tsuyuki, “The use of fixed–wing UAV photogrammetry with LiDAR DTM to estimate merchantable volume and carbon stock in living biomass over a mixed conifer–broadleaf forest,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 73, pp. 767–777, Dec. 2018, doi: https://doi.org/10.1016/j.jag.2018.08.017.

[104] Keqi Zhang, Shu-Ching Chen, D. Whitman, Mei-Ling Shyu, Jianhua Yan, and Chengcui Zhang, “A progressive morphological filter for removing nonground measurements from airborne LIDAR data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 4, pp. 872–882, Apr. 2003, doi: https://doi.org/10.1109/tgrs.2003.810682.

[105] W. Zhang *et al.*, “An Easy-to-Use Airborne LiDAR Data Filtering Method Based on Cloth Simulation,” *Remote Sensing*, vol. 8, no. 6, p. 501, Jun. 2016, doi: https://doi.org/10.3390/rs8060501.

[106] R. Wack and A. Wimmer, “Digital terrain models from airborne laserscanner data-a grid based approach,” *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, vol. 34, Art. no. 3/B, 2002.

[107] T. J. Pingel, K. C. Clarke, and W. H. McBride, “An improved simple morphological filter for the terrain classification of airborne LIDAR data,” *Isprs Journal of Photogrammetry and Remote Sensing*, vol. 77, pp. 21–30, Mar. 2013, doi: https://doi.org/10.1016/j.isprsjprs.2012.12.002.

[108] E. Sanz-Ablanedo, J. Chandler, J. Rodríguez-Pérez, and C. Ordóñez, “Accuracy of Unmanned Aerial Vehicle (UAV) and SfM Photogrammetry Survey as a Function of the Number and Location of Ground Control Points Used,” *Remote Sensing*, vol. 10, no. 10, p. 1606, Oct. 2018, doi: https://doi.org/10.3390/rs10101606.

[109] E. Ferrer-González, F. Agüera-Vega, F. Carvajal-Ramírez, and P. Martínez-Carricondo, “UAV Photogrammetry Accuracy Assessment for Corridor Mapping Based on the Number and Distribution of Ground Control Points,” *Remote Sensing*, vol. 12, no. 15, p. 2447, Jul. 2020, doi: https://doi.org/10.3390/rs12152447.

[110] D. Girardeau-Montaut, “CloudCompare – Open-Source project,” *www.cloudcompare.org*. https://www.cloudcompare.org/

[111] S. Walfish, “A review of statistical outlier methods,” *Pharmaceutical technology*, vol. 30, Art. no. 11, 2006.

[112] Z. Jitao and Z. Tao, “Filtering of airborne LiDAR point cloud based on variable radius circle and B-spline fitting,” *Acta Geodaetica Et Cartographica Sinica*, vol. 44, Art. no. 12, 2015.

Appendix A. Point Cloud Processing and Weed Detection Code (Python)

```
import os
import numpy as np
import open3d as o3d
import myfunc as mf

os.system('cls')

# ------------ USER PATHS------------
UAV_PLY   = r"C:\path\to\uav.ply"
NANO_PLY  = r"C:\path\to\nano.ply"
OUT_DIR   = r"C:\path\to\out"
os.makedirs(OUT_DIR, exist_ok=True)

# ------------ 0) Load ------------
uav  = mf.load_pcd(UAV_PLY)
nano = mf.load_pcd(NANO_PLY)

# ------------ 1) GCP-based initial alignment ------------
# Extract 3 red targets from both clouds
```

```
uav_gcps  = mf.extract_red_gcps_dbscan(uav,  eps=0.08, min_samples=60)
nano_gcps = mf.extract_red_gcps_dbscan(nano, eps=0.08, min_samples=60)

# Order via triangle geometry matching
ref3, tgt3 = mf.match_gcp_order_by_triangle(uav_gcps, nano_gcps)

# Rigid transform nano -> uav
rig = mf.rigid_transform(np.vstack(tgt3), np.vstack(ref3), with_scale=False)
nano_init = mf.apply_transform(nano, rig.R, rig.t, rig.s)

# ------------ 2) ICP refinement ------------
nano_icp, T_icp, reg = mf.icp_refine(nano_init, uav, max_corr_dist=0.20)
aligned = uav + nano_icp   # integrated PCD

mf.save_pcd(aligned, os.path.join(OUT_DIR, "A_aligned.ply"))
print("[SAVE] Integrated aligned cloud -> A_aligned.ply")

# ------------ 3) Common helper to run per-source branch ------------
def branch_process(prefix: str, whole_pcd: o3d.geometry.PointCloud):
    """
    prefix: 'N' for UAV-only, 'A' for integrated
    """
    # 3.1 denoise
    clean = mf.remove_statistical_outliers(whole_pcd, 15, 2.0)
    clean = mf.remove_radius_outliers(clean, radius=0.05, min_points=8)
    mf.save_pcd(clean, os.path.join(OUT_DIR, f"{prefix}_clean.ply"))


    # 3.2 PMF ground/non-ground
    g_path = os.path.join(OUT_DIR, f"{prefix}_ground.ply")
    ng_path= os.path.join(OUT_DIR, f"{prefix}_nonground.ply")
    g_out, ng_out = mf.pmf_filter_file(
        input_path=os.path.join(OUT_DIR, f"{prefix}_clean.ply"),
        ground_out_path=g_path,
        nonground_out_path=ng_path,
        max_window_size=40, slope=1.1, initial_distance=0.9, max_distance=2.8, cell_size=1.2
    )
    # Fallback if PDAL not available: split nothing
    ground_pcd   = mf.load_pcd(g_path)   if g_out  else o3d.geometry.PointCloud()
    nonground_pcd= mf.load_pcd(ng_path)  if ng_out else whole_pcd


    # 3.3 denoise ground before soil-veg split
    if not ground_pcd.is_empty():
        ground_pcd = mf.remove_statistical_outliers(ground_pcd, 15, 2.0)
        ground_pcd = mf.remove_radius_outliers(ground_pcd, 0.05, 8)
        mf.save_pcd(ground_pcd, os.path.join(OUT_DIR, f"{prefix}_ground_d.ply"))

    # 3.4 soil-vegetation KMeans (weights: X,Y,relZ,ExG,ExGR)
    # For relZ plane, we use the UAV ground as reference if available; otherwise the current ground.
    ref_ground = ground_pcd if prefix == "N" else mf.load_pcd(os.path.join(OUT_DIR, "N_ground.ply")) if os.path.exists(os.path.join(OUT_DIR, "N_ground.ply")) else ground_pcd
    _, soil_mask, veg_mask, colored = mf.kmeans_soil_vs_veg(ground_pcd, ref_ground, weights=(0.4,0.4,2.3,2.0,1.0))
    mf.save_pcd(colored, os.path.join(OUT_DIR, f"{prefix}_soilveg_colored.ply"))
    soil = ground_pcd.select_by_index(np.where(soil_mask)[0])
    veg  = ground_pcd.select_by_index(np.where(veg_mask)[0])
    mf.save_pcd(soil, os.path.join(OUT_DIR, f"{prefix}_soil.ply"))
    mf.save_pcd(veg,  os.path.join(OUT_DIR, f"{prefix}_veg.ply"))

    # 3.5 weed candidates (use crop rows from non-ground)
    crop = nonground_pcd
    crop = mf.voxel_downsample(crop, 0.02)
    veg  = mf.voxel_downsample(veg,  0.02)
    # 3.6 cluster weeds (DBSCAN -> KMeans size split -> length filter)
    labels, clusters, centers, colored_db = mf.cluster_weeds_dbscan_xyz(
        veg, eps=0.14, min_samples=18, axis_weights=(1,1,1)
    )
    mf.save_pcd(colored_db, os.path.join(OUT_DIR, f"{prefix}_weed_dbscan_color.ply"))

    keep_ids, _, _ = mf.select_small_clusters_by_kmeans(clusters, k=2)
    keep_ids = mf.enforce_max_cluster_length(keep_ids, clusters, max_len=1.0, mode="xy")

    # Build weeds-only cloud (all red) & multi-color result
    uniq = sorted(set(labels) - {-1})
    lab2idx = {lab: i for i, lab in enumerate(uniq)}
    weed_labels = [lab for lab in uniq if lab2idx[lab] in keep_ids]
    if len(weed_labels) > 0:
        mask_pts = np.isin(labels, weed_labels)
        weeds_only = veg.select_by_index(np.where(mask_pts)[0])
        weeds_only_red = o3d.geometry.PointCloud(weeds_only)
        weeds_only_red.paint_uniform_color([1.0, 0.0, 0.0])
        mf.save_pcd(weeds_only_red, os.path.join(OUT_DIR,
f"{prefix}_weeds_only_red.ply"))


    # 3.7 Weed map (2D overlay)
    # crops = non-ground (green); weeds = weeds_only_red (red)
    crops_for_map = crop
    weeds_for_map = weeds_only_red if len(weed_labels) > 0 else
o3d.geometry.PointCloud()
    mf.make_weed_map(
        crops_for_map, weeds_for_map,
        out_png=os.path.join(OUT_DIR, f"{prefix}_weed_map.png"),
        pixel_size=0.02, margin=0.5, point_dilate=2
    )


# ------------ Run both branches ------------
# UAV-only branch
branch_process("N", uav)


# Integrated (aligned) branch
branch_process("A", aligned)


print("\n[Done] Main pipeline completed.\n")
```

```
# (DBSCAN / KMeans weed post-processing & 2D maps follow your original flow/styles
#  for A_* and N_* branches. )
```

Appendix B. Function Define Code (Python)

```
import json

import os

from dataclasses import dataclass

from typing import List, Tuple, Optional



import numpy as np

import open3d as o3d



# ---------- Basic IO / Viz ----------



def load_pcd(path: str) -> o3d.geometry.PointCloud:

    p = o3d.io.read_point_cloud(path)

    if p.is_empty():

        raise ValueError(f"[load_pcd] Empty or unreadable: {path}")

    return p



def save_pcd(pcd: o3d.geometry.PointCloud, path: str):

    o3d.io.write_point_cloud(path, pcd)



def visualize_pcd(*geoms, window="Open3D", point_size=2.0):
    vis = o3d.visualization.Visualizer()
    vis.create_window(window_name=window)
    for g in geoms:
        vis.add_geometry(g)
    opt = vis.get_render_option()
    opt.point_size = float(point_size)
    vis.run()
    vis.destroy_window()


def voxel_downsample(pcd: o3d.geometry.PointCloud, voxel_size: float) ->
o3d.geometry.PointCloud:
    return pcd.voxel_down_sample(max(voxel_size, 1e-6))


# ---------- Denoising ----------


def remove_statistical_outliers(
    pcd: o3d.geometry.PointCloud, nb_neighbors=15, std_ratio=2.0
) -> o3d.geometry.PointCloud:
    pcd2, _ = pcd.remove_statistical_outlier(nb_neighbors=nb_neighbors,
std_ratio=std_ratio)
    return pcd2
```

```
def remove_radius_outliers(
    pcd: o3d.geometry.PointCloud, radius=0.05, min_points=8
) -> o3d.geometry.PointCloud:
    pcd2, _ = pcd.remove_radius_outlier(nb_points=min_points, radius=radius)
    return pcd2


# ---------- GCP extraction (red targets) ----------


def threshold_red_mask(rgb: np.ndarray, r_min=180, g_max=120, b_max=100) ->
np.ndarray:
    """Return boolean mask for near-red points (colors 0–255 or 0–1 both ok)."""
    C = rgb.copy()
    if C.max() <= 1.0:
        C = (C * 255.0).astype(np.uint8)
    r, g, b = C[:, 0], C[:, 1], C[:, 2]
    return (r >= r_min) & (g <= g_max) & (b <= b_max)


def dbscan_labels(X: np.ndarray, eps: float, min_samples: int) -> np.ndarray:
    from sklearn.cluster import DBSCAN
    return DBSCAN(eps=eps, min_samples=min_samples).fit_predict(X)
```

```
def extract_red_gcps_dbscan(
    pcd: o3d.geometry.PointCloud,
    eps=0.08,
    min_samples=50,
    r_min=180, g_max=120, b_max=100
) -> List[np.ndarray]:
    """Return list of 3D centroids for each red GCP cluster."""
    P = np.asarray(pcd.points)
    C = np.asarray(pcd.colors)
    mask = threshold_red_mask(C, r_min=r_min, g_max=g_max, b_max=b_max)
    if mask.sum() == 0:
        raise ValueError("[GCP] No red-like points found with threshold.")
    Psub = P[mask]
    labels = dbscan_labels(Psub, eps=eps, min_samples=min_samples)
    uniq = [lab for lab in sorted(set(labels)) if lab != -1]
    if len(uniq) < 3:
        raise ValueError(f"[GCP] Only {len(uniq)} red clusters found; need ⩾3.")
    centroids = [Psub[labels == lab].mean(axis=0) for lab in uniq]
    return centroids  # unsorted; sort next step


# ---------- GCP triangle-order matching ----------
```

```
def _triangle_side_lengths(pts: np.ndarray) -> np.ndarray:

    a, b, c = pts

    return np.array([

        np.linalg.norm(a - b),

        np.linalg.norm(b - c),

        np.linalg.norm(c - a)

    ])


def match_gcp_order_by_triangle(

    ref_pts3: List[np.ndarray], tgt_pts3: List[np.ndarray]

) -> Tuple[np.ndarray, np.ndarray]:

    """Match target GCP order to reference by triangle geometry (shape-invariant)."""

    from itertools import permutations

    ref = np.vstack(ref_pts3)[:3]

    best = None

    best_diff = 1e18

    ref_len = _triangle_side_lengths(ref)

    for perm in permutations(np.vstack(tgt_pts3)[:3]):

        tgt = np.vstack(perm)

        diff = np.linalg.norm(np.sort(ref_len) - np.sort(_triangle_side_lengths(tgt)))
        if diff < best_diff:
            best_diff = diff
            best = tgt
    return ref, best


# ---------- Rigid transform (Procrustes/Kabsch) ----------


@dataclass
class RigidResult:
    R: np.ndarray  # (3,3)
    t: np.ndarray  # (3,)
    s: float       # scale (1.0 for rigid)


def rigid_transform(A: np.ndarray, B: np.ndarray, with_scale=False) -> RigidResult:
    """Compute transform that maps A -> B (rows are points)."""
    assert A.shape == B.shape and A.shape[1] == 3
    muA = A.mean(axis=0)
    muB = B.mean(axis=0)
    AA = A - muA
    BB = B - muB
    H = AA.T @ BB
    U, S, Vt = np.linalg.svd(H)

    R = Vt.T @ U.T

    if np.linalg.det(R) < 0:

        Vt[-1, :] *= -1

        R = Vt.T @ U.T

    if with_scale:

        varA = (AA**2).sum()

        s = (S.sum() / varA) if varA > 1e-12 else 1.0

    else:

        s = 1.0

    t = muB - s * (R @ muA)

    return RigidResult(R=R, t=t, s=s)


def apply_transform(pcd: o3d.geometry.PointCloud, R: np.ndarray, t: np.ndarray, s:
float=1.0):

    P = np.asarray(pcd.points)

    P2 = (s * (R @ P.T)).T + t

    out = o3d.geometry.PointCloud(pcd)

    out.points = o3d.utility.Vector3dVector(P2)

    return out
```

```
# ---------- ICP refinement ----------

def icp_refine(
    source: o3d.geometry.PointCloud,
    target: o3d.geometry.PointCloud,
    max_corr_dist=0.2,
    init=o3d.geometry.TransformationEstimationPointToPoint()
):
    trans_init = np.eye(4)
    reg = o3d.pipelines.registration.registration_icp(
        source, target, max_corr_dist, trans_init,
        o3d.pipelines.registration.TransformationEstimationPointToPoint()
    )
    T = reg.transformation
    src2 = o3d.geometry.PointCloud(source)
    src2.transform(T)
    return src2, T, reg

# ---------- PMF ground filtering (PDAL wrapper with fallback) ----------

def pmf_filter_file(input_path: str,
                    ground_out_path: str,
                    nonground_out_path: str,
                    max_window_size=40, slope=1.1,
                    initial_distance=0.9, max_distance=2.8,
                    cell_size=1.2) -> Tuple[Optional[str], Optional[str]]:
    """
    Try PDAL PMF via subprocess. If PDAL not available, returns (None, None).
    """
    import subprocess, tempfile
    if not os.path.exists(input_path):
        raise FileNotFoundError(input_path)

    # PDAL pipeline for PMF on generic PLY (reads XYZ, optional RGB)
    pipe = {
        "pipeline": [
            {"type": "readers.ply", "filename": input_path},
            {"type": "filters.pmf",
             "max_window_size": max_window_size,
             "slope": slope,
             "initial_distance": initial_distance,
             "max_distance": max_distance,
             "cell_size": cell_size},
            {"type": "filters.range", "limits": "Classification[2:2]"},
            {"type": "writers.ply", "filename": ground_out_path, "storage_mode": "ascii"},
        ]
    }
    pipe2 = {
        "pipeline": [
            {"type": "readers.ply", "filename": input_path},
            {"type": "filters.pmf",
             "max_window_size": max_window_size,
             "slope": slope,
             "initial_distance": initial_distance,
             "max_distance": max_distance,
             "cell_size": cell_size},
            {"type": "filters.range", "limits": "Classification[2:2]"},
            {"type": "filters.assign", "assignment": "Classification[:]=0"},
            {"type": "filters.merge"},
            {"type": "writers.ply", "filename": nonground_out_path, "storage_mode": "ascii"}
        ]
    }
    try:
        with tempfile.NamedTemporaryFile(mode="w", suffix=".json", delete=False) as f1:
            json.dump(pipe, f1); p1 = f1.name
        with tempfile.NamedTemporaryFile(mode="w", suffix=".json", delete=False) as f2:
            json.dump(pipe2, f2); p2 = f2.name
        subprocess.run(["pdal", "pipeline", p1], check=True)
        subprocess.run(["pdal", "pipeline", p2], check=True)
        return ground_out_path, nonground_out_path
    except Exception as e:
        print("[PMF] PDAL not available or failed:", e)
        return None, None


# ---------- Vegetation indices / features ----------


def compute_ExG(colors01: np.ndarray) -> np.ndarray:
    C = colors01
    if C.max() <= 1.0:
        r, g, b = C[:, 0], C[:, 1], C[:, 2]
    else:
        r, g, b = C[:, 0]/255.0, C[:, 1]/255.0, C[:, 2]/255.0
    return 2*g - r - b
```

```
def compute_ExGR(colors01: np.ndarray) -> np.ndarray:

    C = colors01

    if C.max() <= 1.0:

        r, g, b = C[:, 0], C[:, 1], C[:, 2]

    else:

        r, g, b = C[:, 0]/255.0, C[:, 1]/255.0, C[:, 2]/255.0

    exg = 2*g - r - b

    exr = 1.4*r - g

    return exg - exr


def fit_plane_svd(points_xyz: np.ndarray):

    P = np.asarray(points_xyz)

    c = P.mean(axis=0)

    U, S, Vt = np.linalg.svd(P - c, full_matrices=False)

    n = Vt[-1, :]

    n = n / (np.linalg.norm(n) + 1e-12)

    a, b, c0 = n

    d = -np.dot(n, c)

    return (a, b, c0, d), n


def relative_z_from_plane(points_xyz: np.ndarray, plane):
    a, b, c, d = plane

    P = np.asarray(points_xyz)

    z_plane = (-d - a*P[:,0] - b*P[:,1]) / (c + 1e-12)

    return P[:,2] - z_plane


# ---------- Soil vs vegetation (weighted K-means) ----------


def kmeans_soil_vs_veg(

    ground_pcd: o3d.geometry.PointCloud,

    ref_ground: o3d.geometry.PointCloud,

    weights=(0.4, 0.4, 2.3, 2.0, 1.0),

    k_nn_ref=3,

    random_state=42

):

    """

    Features: [X, Y, relZ, ExG, ExGR]. relZ derived from plane fit to ref_ground.

    Returns: labels, soil_mask, veg_mask, colored_copy

    """

    from sklearn.cluster import KMeans


    P = np.asarray(ground_pcd.points)
    C = np.asarray(ground_pcd.colors) if ground_pcd.has_colors() else np.zeros((len(P), 3))
    # fit plane on reference ground for robust relZ
    plane, _ = fit_plane_svd(np.asarray(ref_ground.points))
    relZ = relative_z_from_plane(P, plane)
    ExG = compute_ExG(C); ExGR = compute_ExGR(C)

    X = np.column_stack([P[:,0], P[:,1], relZ, ExG, ExGR])
    W = np.asarray(weights, dtype=float)
    Xw = X * W  # simple diagonal weighting

    km = KMeans(n_clusters=2, n_init=10, random_state=random_state).fit(Xw)
    labels = km.labels_

    # Heuristic: soil has lower ExG/ExGR and lower relZ
    # Determine which cluster is soil by comparing mean (relZ + ExG)
    c0 = X[labels==0]; c1 = X[labels==1]
    score0 = c0[:,2].mean() + c0[:,3].mean()
    score1 = c1[:,2].mean() + c1[:,3].mean()
    soil_label = 0 if score0 < score1 else 1
    soil_mask = (labels == soil_label)
    veg_mask  = ~soil_mask

    # Build colored copy for quick view (brown soil, green veg)
    out = o3d.geometry.PointCloud(ground_pcd)
    col = np.tile([0.6, 0.4, 0.2], (len(P), 1))
    col[veg_mask] = [0.0, 0.8, 0.0]
    out.colors = o3d.utility.Vector3dVector(col)

    return labels, soil_mask, veg_mask, out
# (Adapted to your weights/feature-set used in your scripts )

# ---------- Weed clustering & filters ----------

def cluster_weeds_dbscan_xyz(
    weeds_cand: o3d.geometry.PointCloud, eps=0.14, min_samples=18, axis_weights=(1,1,1)
):
    """
    DBSCAN in XYZ with optional anisotropic weights.
    Returns labels, clusters(list of Nx3 arrays), centers, colored_cloud
    """
    from sklearn.cluster import DBSCAN
    P = np.asarray(weeds_cand.points)
    W = np.array(axis_weights, dtype=float)
    X = P * W
    labels = DBSCAN(eps=eps, min_samples=min_samples).fit_predict(X)
    uniq = [lab for lab in sorted(set(labels)) if lab != -1]
    clusters = [P[labels == lab] for lab in uniq]
    centers = [c.mean(axis=0) for c in clusters]

    # colored cloud for preview: each cluster random color, noise gray
    colored = o3d.geometry.PointCloud(weeds_cand)
    C = np.tile([0.8, 0.8, 0.8], (len(P), 1))
    rng = np.random.default_rng(2025)
    for lab in uniq:
        color = rng.random(3) * 0.6 + 0.35
        C[labels == lab] = color
    colored.colors = o3d.utility.Vector3dVector(C)
    return labels, clusters, centers, colored
# (Colorization / reporting style follows your main DBSCAN pipelines )


def select_small_clusters_by_kmeans(clusters: List[np.ndarray], k=2):
    """Split clusters by size (max span in XY) via KMeans; return indices of smaller group."""
    from sklearn.cluster import KMeans
    spans = []
    for c in clusters:
        if len(c) == 0:
            spans.append(0.0)
            continue
        xy = c[:, :2]
        ext = xy.max(axis=0) - xy.min(axis=0)
        spans.append(float(np.linalg.norm(ext)))
    spans = np.array(spans).reshape(-1, 1)
    km = KMeans(n_clusters=k, n_init=10, random_state=42).fit(spans)
    labs = km.labels_
    means = [spans[labs == i].mean() if (labs == i).any() else 0 for i in range(k)]
    small_id = int(np.argmin(means))
    keep_ids = list(np.where(labs == small_id)[0])
    return keep_ids, labs, km.cluster_centers_


def enforce_max_cluster_length(keep_ids: List[int], clusters: List[np.ndarray], max_len=1.0, mode="xy"):
    """Drop clusters whose maximum extent exceeds threshold (m)."""
    out = []
    for idx in keep_ids:
        c = clusters[idx]
        if len(c) < 2:
            continue
        if mode.lower() == "xy":
            u = c[:, :2]
        else:
            u = c
        ext = u.max(axis=0) - u.min(axis=0)
        L = float(np.linalg.norm(ext))
        if L <= max_len:
            out.append(idx)
    return out


# ---------- 2D maps (crop-only, crop+weed) ----------


def make_crop_map(crop_pcd: o3d.geometry.PointCloud, out_png: str,
                  pixel_size=0.02, margin=0.5, point_dilate=2):
    """
    Top-down 2D projection of crop (green); adapted from your helper script.
```

```
(Original idea/canvas math mirrored here) :contentReference[oaicite:4]{index=4}
"""
P = np.asarray(crop_pcd.points)[:, :2]
min_xy = P.min(axis=0) - margin
max_xy = P.max(axis=0) + margin
size_xy = max_xy - min_xy
W = max(int(np.ceil(size_xy[0] / pixel_size)), 1)
H = max(int(np.ceil(size_xy[1] / pixel_size)), 1)
img = np.full((H, W, 3), 255, dtype=np.uint8)

def xy_to_rc(xy):
    cols = ((xy[:, 0] - min_xy[0]) / pixel_size).astype(int)
    rows = (H - 1 - (xy[:, 1] - min_xy[1]) / pixel_size).astype(int)
    return np.clip(rows, 0, H-1), np.clip(cols, 0, W-1)

r, c = xy_to_rc(P)
img[r, c] = np.array([0, 200, 0], dtype=np.uint8)
try:
    from PIL import Image
    Image.fromarray(img).save(out_png)
except Exception:
        import matplotlib.pyplot as plt

        plt.imsave(out_png, img)

    return img, (min_xy, max_xy), (H, W)


def make_weed_map(crop_pcd, weed_pcd, out_png, pixel_size=0.02, margin=0.5,
point_dilate=2):

    """

    2D projection: crops green + weeds red (overlaid).

    Adapted from your weedmap helper (canvas bounds + paint)
:contentReference[oaicite:5]{index=5}

    """

    pts = []

    if not crop_pcd.is_empty(): pts.append(np.asarray(crop_pcd.points)[:, :2])

    if not weed_pcd.is_empty(): pts.append(np.asarray(weed_pcd.points)[:, :2])

    XY = np.vstack(pts)

    min_xy = XY.min(axis=0) - margin

    max_xy = XY.max(axis=0) + margin

    size_xy = max_xy - min_xy

    W = max(int(np.ceil(size_xy[0] / pixel_size)), 1)

    H = max(int(np.ceil(size_xy[1] / pixel_size)), 1)

    img = np.full((H, W, 3), 255, dtype=np.uint8)
```

```
def xy_to_rc(xy):
    cols = ((xy[:, 0] - min_xy[0]) / pixel_size).astype(int)
    rows = (H - 1 - (xy[:, 1] - min_xy[1]) / pixel_size).astype(int)
    return np.clip(rows, 0, H-1), np.clip(cols, 0, W-1)

if not crop_pcd.is_empty():
    r, c = xy_to_rc(np.asarray(crop_pcd.points)[:, :2])
    img[r, c] = np.array([0, 200, 0], dtype=np.uint8)
if not weed_pcd.is_empty():
    r, c = xy_to_rc(np.asarray(weed_pcd.points)[:, :2])
    img[r, c] = np.array([220, 20, 60], dtype=np.uint8)

try:
    from PIL import Image
    Image.fromarray(img).save(out_png)
except Exception:
    import matplotlib.pyplot as plt
    plt.imsave(out_png, img)
return img, (min_xy, max_xy), (H, W)
```

Appendix C. Point Cloud Evaluation Code (Python)

```
import open3d as o3d

import numpy as np

import pandas as pd



# Optional: convex hull area via SciPy (better than AABB). If SciPy isn't available,

# the code falls back to AABB area automatically.

try:

    from scipy.spatial import ConvexHull

    SCIPY_OK = True

except Exception:

    SCIPY_OK = False



def read_pcd(path):

    p = o3d.io.read_point_cloud(path)

    assert len(p.points) > 0, f"Empty point cloud: {path}"

    return p



def fit_plane_svd(points_xyz):

    """Least-squares plane fit: returns (a,b,c,d) for ax+by+cz+d=0 and unit normal."""

    P = np.asarray(points_xyz)
    centroid = P.mean(axis=0)
    U, S, Vt = np.linalg.svd(P - centroid, full_matrices=False)
    normal = Vt[-1, :]
    normal = normal / (np.linalg.norm(normal) + 1e-12)
    a, b, c = normal
    d = -centroid @ normal
    return (a, b, c, d), normal, centroid


def relative_z_from_plane(points_xyz, plane):
    """Relative height above the plane evaluated at the XY of each point."""
    a, b, c, d = plane
    P = np.asarray(points_xyz)
    # signed distance to plane
    dist = (a*P[:,0] + b*P[:,1] + c*P[:,2] + d) / (np.sqrt(a*a+b*b+c*c) + 1e-12)
    # If you prefer purely vertical height (z minus plane's z at (x,y)), use this instead:
    # z_plane = ( -d - a*P[:,0] - b*P[:,1]) / (c + 1e-12)
    # dist = P[:,2] - z_plane
    return dist


def ground_area_m2(points_xyz, method="convex_hull_or_aabb", cell=None):
    """Approximate ground-projected area (m²). Uses convex hull in XY if SciPy is present; else AABB.
       If 'cell' is provided (e.g., 0.1 m), returns raster area as nx*ny*cell^2 (optional)."""
    XY = np.asarray(points_xyz)[:, :2]
    if cell is not None:
        xy_min = XY.min(axis=0); xy_max = XY.max(axis=0)
        nx, ny = np.ceil((xy_max-xy_min)/cell).astype(int)
        return float(nx*ny*(cell**2))
    if SCIPY_OK:
        hull = ConvexHull(XY)
        return float(hull.area)  # in 2D, ConvexHull.area gives perimeter; use volume for area
    # SciPy quirk: in 2D, 'volume' is the polygon area
    if SCIPY_OK:
        return float(ConvexHull(XY).volume)
    # Fallback: AABB area
    mins = XY.min(axis=0); maxs = XY.max(axis=0)
    ext = maxs - mins
    return float(ext[0]*ext[1])


def average_density_pts_per_m2(pcd, area_m2=None):
    if area_m2 is None:
        area_m2 = ground_area_m2(pcd.points)
    return len(pcd.points) / max(area_m2, 1e-9)


def percentage_ground_level(relative_z, band=(0.0, 0.03)):
    """Percent of points within a small band above ground (tune band to your crop)."""
    lo, hi = band
    mask = (relative_z >= lo) & (relative_z <= hi)
    return mask.mean() * 100.0


# Replace with your actual file paths
UAV_PLY      = r"C:\path\to\uav.ply"
NANO_PLY     = r"C:\path\to\nano.ply"
ALIGNED_PLY  = r"C:\path\to\aligned.ply"
GROUND_PLY   = r"C:\path\to\uav_ground.ply"  # PMF output


pcd_uav     = read_pcd(UAV_PLY)
pcd_nano    = read_pcd(NANO_PLY)
pcd_aligned = read_pcd(ALIGNED_PLY)
pcd_ground  = read_pcd(GROUND_PLY)


# Fit plane to the ground points
```

```
plane, normal, centroid = fit_plane_svd(pcd_ground.points)


# Compute relative z for each dataset
rz_uav     = relative_z_from_plane(pcd_uav.points, plane)
rz_nano    = relative_z_from_plane(pcd_nano.points, plane)
rz_aligned = relative_z_from_plane(pcd_aligned.points, plane)


area_uav     = ground_area_m2(pcd_uav.points)
area_nano    = ground_area_m2(pcd_nano.points)
area_aligned = ground_area_m2(pcd_aligned.points)


dens_uav     = average_density_pts_per_m2(pcd_uav, area_uav)
dens_nano    = average_density_pts_per_m2(pcd_nano, area_nano)
dens_aligned = average_density_pts_per_m2(pcd_aligned, area_aligned)


summary_density = pd.DataFrame({
    "Cloud": ["UAV","Nano-drone","Aligned"],
    "Points (N)": [len(pcd_uav.points), len(pcd_nano.points), len(pcd_aligned.points)],
    "Area (m²)": [area_uav, area_nano, area_aligned],
    "Avg density (pts/m²)": [dens_uav, dens_nano, dens_aligned],
})
```

```
print(summary_density.round(3))


# Define your "ground-level" band (meters above the plane)
BAND = (0, 0.5)  # 0–3 cm; adjust to your site


p_ground_uav     = percentage_ground_level(rz_uav, BAND)
p_ground_nano    = percentage_ground_level(rz_nano, BAND)
p_ground_aligned = percentage_ground_level(rz_aligned, BAND)


summary_ground = pd.DataFrame({
    "Cloud": ["UAV","Nano-drone","Aligned"],
    "Ground-level band (m)": [f"{BAND[0]}–{BAND[1]}"]*3,
    "Ground-level (%)": [p_ground_uav, p_ground_nano, p_ground_aligned],
})
print(summary_ground.round(2))


import matplotlib.pyplot as plt


# Choose consistent bins for comparability (e.g., -0.05 to 2.0 m in 0.05 m steps)
bins = np.arange(-0.05, 2.05, 0.05)
```

```
def plot_hist(rz, title, bins, savepath=None):
    plt.figure(figsize=(6,4))
    plt.hist(rz, bins=bins)
    plt.xlabel("Relative z (m)")
    plt.ylabel("Count")
    plt.title(title)
    plt.tight_layout()
    if savepath:
        plt.savefig(savepath, dpi=300)
    plt.show()


plot_hist(rz_uav,     "UAV — Relative z distribution", bins, "hist_uav.png")
plot_hist(rz_nano,    "Nano-drone — Relative z distribution", bins, "hist_nano.png")
plot_hist(rz_aligned, "Aligned — Relative z distribution", bins, "hist_aligned.png")


# Optional: export the histogram counts for your thesis appendix
def hist_to_df(rz, bins, name):
    h, e = np.histogram(rz, bins=bins)
    centers = 0.5*(e[:-1]+e[1:])
    return pd.DataFrame({"Cloud":name, "RelZ_center_m":centers, "Count":h})
```

```
hist_df = pd.concat([

    hist_to_df(rz_uav, bins, "UAV"),

    hist_to_df(rz_nano, bins, "Nano-drone"),

    hist_to_df(rz_aligned, bins, "Aligned"),

], ignore_index=True)

hist_df.to_csv("relative_z_histograms.csv", index=False)
```