2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Temporal- and Viewpoint-Invariant Registration for Under-Canopy Footage using Deep-Learning-based Bird's-Eye View Prediction

Jiawei Zhou[1], Ruben Mascaro[1], Cesar Cadena[2], Margarita Chli[1], Lucas Teixeira[1]
[1] Vision for Robotics Lab - ETH Zurich and University of Cyprus,
[2] Robotic Systems Lab - ETH Zurich

***Abstract*— Conducting visual assessments under the canopy using mobile robots is an emerging task in smart farming and forestry. However, it is challenging to register images across different data-collection days, especially across seasons, due to the self-occluding geometry and temporal dynamics in forests and orchards. This paper proposes a new approach for registering under-canopy image sequences in general and in these situations. Our methodology leverages standard GPS data and deep-learning-based perspective to bird's-eye view conversion to provide an initial estimation of the positions of the trees in images and their association across datasets. Furthermore, it introduces an innovative strategy for extracting tree trunks and clean ground surfaces from noisy and sparse 3D reconstructions created from the image sequences, utilizing these features to achieve precise alignment. Our robust alignment method effectively mitigates position and scale drift, which may arise from GPS inaccuracies and Sparse Structure from Motion (SfM) limitations. We evaluate our approach on three challenging real-world datasets, demonstrating that our method outperforms ICP-based methods on average by 50%, and surpasses FGR and TEASER++ by over 90% in alignment accuracy. These results highlight our method's cost efficiency and robustness, even in the presence of severe outliers and sparsity. https://github.com/VIS4ROB-lab/bev_undercanopy_registration**

I. Introduction

Visual inspection plays an increasingly vital role in modern orchards and forest management. This practice not only facilitates the continuous monitoring of plant production and health conditions [1] but also enables per-plant decision-making, such as determining the optimal amounts of agricultural chemicals to be applied to individual trees, a task that can nowadays be executed by autonomous robots [2]. In this context, achieving precise registration between successive data capture sessions becomes essential. This is particularly challenging for under-canopy footage due to the repetitive nature of leaves, self-occluding geometry, and dramatic changes in the capturable features of such environments. As shown in Fig. 1, the difficulty in finding feature correspondences across sessions due to partial views and temporal changes, including seasonal changes, growth of vegetation, and weather conditions, poses a major problem. Furthermore, the task is complicated by the need to deal with varying light conditions and dynamic elements such as moving foliage and shadows.

This work has been partly funded by the European Research Council (ERC), as part of the project SkEyes (Grant agreement no. 101089328), ETH Zurich Research Grant No. 21-1 ETH-27, and by Unity Technologies.

(a) Winter. (b) Summer.

Fig. 1: Challenges for registration of visual datasets with temporal and viewpoint changes.

While numerous strategies for point-cloud registration have been proposed in the literature, ranging from the Iterative Closest Point (ICP) algorithm [3] and its variants [4], [5], [6] to recent deep-learning-based approaches [7], [8], [9], these are typically designed for static environments and are quite sensitive to noise and outliers. Consequently, state-of-the-art methods for point-cloud registration in natural environments aim to first extract time-invariant scene elements, such as tree stems, and then proceed with the alignment based on these stationary landmarks [10], [11], [12]. Still, most studies rely on the availability of precise point clouds provided by terrestrial or aerial laser scanning techniques, which might face limitations in cost and scalability across large areas like orchards. RGB cameras, on the other hand, offer a cost-effective alternative for data collection but lead to noisier 3D reconstructions, complicating subsequent feature extraction and alignment. Furthermore, the likely presence of non-negligible drift within 3D models reconstructed from long image sequences might limit the accuracy of rigid registration processes.

Aiming at addressing these issues, in this paper we present an effective strategy for aligning noisy, sparse point clouds of tree-covered areas reconstructed from RGB images and their GPS positions. By extracting distinct, stable features and performing multi-level reliable alignment based on those, our method can handle the challenges posed by environmental variations effectively and significantly improves registration accuracy across different scanning sessions, mitigating GPS and SfM limitations. In brief, the contributions of this work are the following:

- *Feature extraction for enhanced representation*: We develop a novel method for extracting distinct, representative features from point clouds of tree-covered sites. Specifically, we leverage a learning-based approach [13]

for transforming front-view images of trees into bird's-eye views and roughly estimating the positions of the trees. This information is then used to compute the centers of the tree trunks and isolate the ground points for subsequent model alignment.
- *Iterative alignment of ground and tree centers*: We introduce an iterative alignment process that alternates focus between the segmented ground and the estimated tree centers. This strategy exploits the complementary nature of the extracted features to improve the overall alignment accuracy.
- *Batch-wise model segmentation and local refinement*: We propose a strategic division of the 3D models into several batches, followed by a local refinement phase to fine-tune the alignment results for each batch. This approach mitigates the effect of accumulated errors in the 3D reconstruction and addresses scalability challenges, leading to enhanced alignment precision by focusing on localized feature consistency.

II. Related Work

Point cloud registration is a long-standing problem in Computer Vision and Robotics. This section summarizes recent advances in the field, which can be broadly categorized into optimization-based and deep learning-based [14], and provides an overview of current techniques specifically tailored to point clouds of natural environments.

A. Optimization-based Alignment Methods

Optimization-based methods typically align multiple point clouds by iteratively finding local correspondences and estimating the resulting transformation matrix or by adopting global strategies that search for correspondences across the entire dataset once.

The most emblematic technique within this category is the Iterative Closest Point (ICP) [3] algorithm, which iteratively refines the alignment by minimizing the distance between nearest points. However, ICP has slow convergence and is sensitive to outliers, occlusion, and the initial transformation, often leading to incorrect local minima. Aiming at increasing its robustness and convergence speed, numerous studies have proposed enhancements to the original ICP algorithm over the past years [4], [5], [6]. Recent works have also explored the use of geometric 3D local feature descriptors within ICP-based registration pipelines [15], [16], improving robustness even further. Despite these advancements, challenges persist, particularly regarding the reliance on an initial coarse alignment and the susceptibility to outliers or partial overlaps. These issues become accentuated when dealing with 3D reconstructed natural environments, where the scenes' complexity and noise can severely impact performance.

Beyond ICP, probabilistic-based methods stand out by modeling point clouds as probabilistic distributions. Notable techniques within this domain include Gaussian Mixture Model (GMM)-based methods [17], [18], [19] and Normal Distributions Transform (NDT)-based approaches [20], [21]. These methods handle noise and outliers well with probabilistic models, but distribution parameters may be inconsistent across views, leading only to local convergence. On the other hand, semi-definite programming strategies, such as SDRSAC [22] and TEASER++ [23], frame the estimation of the correspondence assignment matrix as a semi-definite optimization problem, offering a good approximation to the global solution. However, both their efficiency and the size of point clouds they can register remain limited. Other global alignment methods, exemplified by Fast Global Registration (FGR) [24], overcome the dependency on initial alignment by leveraging robust optimization frameworks that can handle noise and sparse data effectively, but are highly dependent on the quality of features. Furthermore, despite their versatility and widespread adoption across various applications, these categories of methods tend to focus on static objects or urban scenarios, struggling with the dynamic and repetitive structures often encountered in natural environments.

B. Deep Learning-based Alignment Methods

Deep-learning-based methods have revolutionized the field of point cloud alignment by utilizing the power of neural networks to learn complex patterns in 3D. These methods can be divided into feature learning and end-to-end learning.

Feature learning methods [7], [25], [26] leverage neural network architectures to extract features and estimate correspondences from point clouds. Generally, their main objective is to assist traditional alignment methods by providing enhanced feature matching. End-to-end alignment models [27], [9], on the other hand, take two point clouds as input and directly predict a transformation matrix for alignment.

In general, deep learning approaches have shown unprecedented performance at handling the complexity and variability of real-world data. However, they come with computational efficiency costs and, currently, the lack of large-scale training datasets capturing the enormous variability of natural environments inhibits their applicability to our particular use-case.

C. Specific Challenges and Solutions for Tree-Covered Sites

Aligning point clouds of tree-covered areas presents unique challenges, usually caused by the presence of significant occlusions, heterogeneous density distributions, and the complex, dynamic structures inherent to natural landscapes. To address these issues, registration methods tailored to this particular problem typically employ advanced feature extraction techniques and leverage distinct environmental landmarks, such as tree trunks, for model alignment. To this end, adaptive strategies are often adopted across both optimization-based and deep-learning frameworks. For instance, Dai et al. [10] identify tree stems through density-based mode detection and subsequently use those to estimate an alignment transformation. RegisMUF [12] utilizes RANSAC for stem extraction followed by guided scanning networks for pairwise registration. Wang et al. [11] apply

cylinder fitting to locate tree stems and use their spatial relationships to estimate the registration transformation. Similarly, Ghorbani et al. [28] implement an aerial-ground registration scheme leveraging tree positions. However, most existing approaches are designed to operate with clean and precise LiDAR point clouds and have not yet been demonstrated on noisier, more challenging vision-based data.

Our research bridges this gap by introducing new robust correspondence finding and alignment methodologies. In particular, we design an approach that uses perspective to bird's-eye view conversion to estimate the positions of trees, targets the extraction of clean ground points and trunks, and employs an iterative alignment process that leverages these complementary features. By addressing critical weaknesses of current techniques and allowing for higher noise levels in the input point clouds, our work pushes forward the state of the art in this field.

III. Method

Our method is designed to register consecutive under-canopy image sequences taken in the same location over months or years from arbitrary directions. We assume that the initial image sequence, which we refer to as the *reference image sequence*, contains either ground-control points (GCPs) or centimeter-level high-precision GNSS/GPS data (i.e., PPK- or RTK-GPS). This allows us to create a sparse 3D model that is both metrically accurate and aligned with gravity. Depending on the environment, a subset of tree locations is also necessary. Subsequent image sequences, which we refer to as *query sequences*, only need traditional GPS data with errors in the range of meters.

An overview of the proposed pipeline is shown in Fig. 2. In the initial stage, a sparse point cloud is computed from the reference image sequence using Structure from Motion (SfM). The result is one or more 3D models, each comprising a set of estimated camera poses and the reconstructed sparse point cloud (note that multiple models are created if SfM cannot find geometrically consistent 2D features between subsets of the input image sequence). Then, these 3D sub-models are merged into a single 3D reference model using PPK-GPS or GCPs. The computed camera poses, the corresponding images, and the given tree positions are used to train a Perspective View to Bird's-Eye View (PV2BEV) network if a model for that crop is unavailable. Finally, temporal-invariant features, namely tree centers and ground points, are extracted from the reference sparse map in the Feature Stage.

The process for the alignment of a query image sequence also starts with a sparse 3D reconstruction, which undergoes the same feature extraction stage as the reference model. Then, the extracted features from both the query and the reference models are used to compute the registration transformation in the Alignment Stage. If sub-model splitting happens during SfM, each sub-model is aligned to the reference model individually due to the lack of high-accuracy GPS or GCPs. The rest of this section explains each of the aforementioned steps in more detail.

A. Geo-referenced Sparse Reconstruction

For each image sequence, a sparse point cloud is initially reconstructed via SfM using COLMAP [29], a widely recognized open-source SfM pipeline known for its robust and well-established performance in feature matching and 3D reconstruction. Other alternatives, such as openMVG [30] and the commercial Pix4D Mapper, would also work. The camera trajectory estimated by COLMAP is then aligned with the available GPS data. However, SfM might lead to inaccuracies in camera pose estimation or scale discrepancies between different parts of a 3D reconstruction due to severe matching errors, especially when dealing with large-scale, complex natural environments. To address these potential issues, we calculate the relative distance between consecutive camera poses in the GPS-aligned 3D reconstruction and compare it to the relative distance between their corresponding GPS positions. This allows us to split the original image sequence into multiple subsequences. Finally, a sparse 3D reconstruction is computed for each subsequence and aligned again with the corresponding GPS data, thereby reducing inaccuracies caused by the initial SfM pass.

B. Training a PV2BEV: Perspective View to Bird's-eye View

PV2BEV networks are designed to receive a frontal image and predict the corresponding top view. In our particular application, we aim to use a PV2BEV model to roughly predict the positions of the visible trees in an image. To this end, and given the lack of under-canopy image samples in common

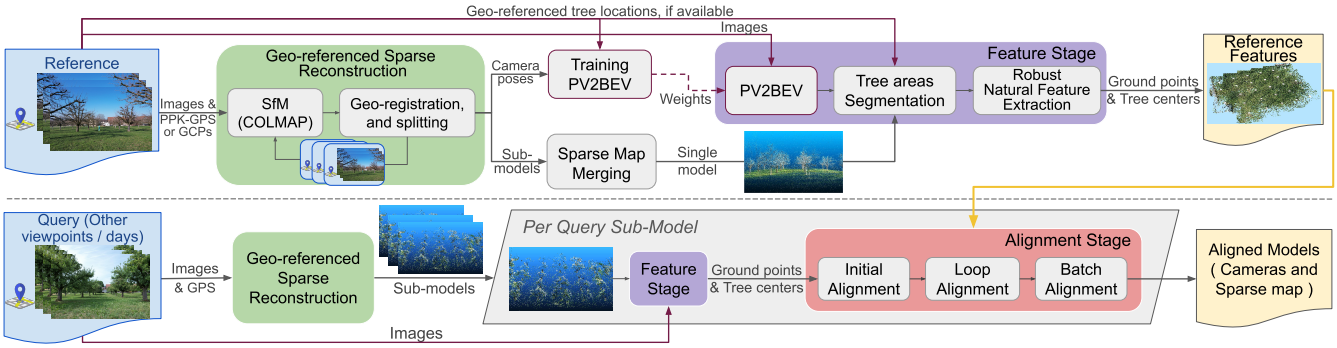


Fig. 2: Overview of the pipeline.

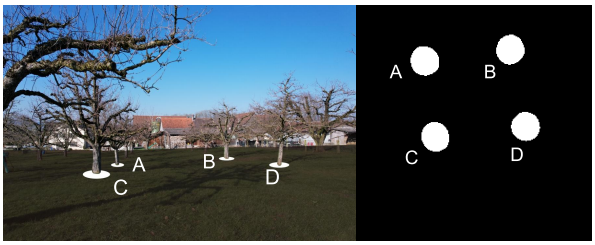


Fig. 3: The top view data predicted from the frontal image. Each white blob represents one of the four closest trees.

training datasets, we train our own model using the reference image sequence and the known tree positions. Figure 3 shows the input and the output of our PV2BEV network. Based on previous work [13], we have chosen to use the *Projecting Your View Attentively* (PYVA) architecture [31] due to its superior performance, but any other PV2BEV network would also be suitable.

During deployment, each time an image is processed, we extract the blob features from the predicted BEV and transform the pixel coordinates into 3D points using the camera poses estimated by SfM. This provides us with approximate initial positions for the trees.

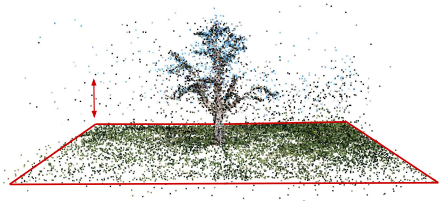
C. Feature Stage

This stage involves the precise identification of distinctive features to be employed for subsequent point-cloud alignment. Particularly, we focus on the segmentation and extraction of temporal-invariant elements, such as ground points and tree centers, that characterize the tree-covered environments targeted in this work.

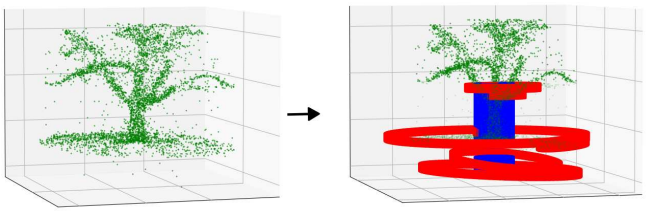
1) Tree Area Segmentation: Given a point cloud representing a portion of a tree-covered environment, the process begins with the segmentation of an area around each tree. This step utilizes the tree position points derived from BEV conversion, with each point acting as the center for segmenting a surrounding area of dimensions $a \times b$. The length a and width b of these segments are determined based on the specific characteristics of the dataset, aiming to ensure that each segment includes a single tree and that, collectively, they encompass the majority of the input point cloud. Fig. 4a illustrates the segmentation process for a single tree area. This preparatory step allows us to concentrate on one tree area at a time, thereby facilitating precise trunk extraction.

2) Robust Natural Feature Extraction: After the initial segmentation, we undertake the extraction of trunk and ground points for each individual tree segment. This process begins with a *pre-processing step* where we segment a cylindrical area around the predicted tree position, thereby mitigating the impact of outliers located outside this predefined zone. To further reduce outliers within this cylindrical region, we employ the DBSCAN algorithm [32], a density-based clustering method known for its effectiveness in outlier detection and removal. This pre-processing step sets the stage for a better extraction of the trunk.

The methodology then progresses with a *slicing and Gaussian fitting step*, executed from the bottom upwards. The



(a) Tree area segmentation.



(b) Slicing and Gaussian fitting (the red cylinder: fitted gaussian; the blue cylinder: expected gaussian).

Fig. 4: The process of tree area segmentation and trunk extraction for a single tree.

process is illustrated in Fig. 4b. Starting at the base, the cylindrical segment is partitioned into multiple horizontal slices. A single Gaussian is applied to model the point distributions within each of those in the X-Y plane. To decide whether each slice belongs to the trunk or not, we first verify whether the covariance of the Gaussian-fitted model falls within a predefined tolerance range of the expected trunk covariance. This step is crucial for distinguishing trunk segments from the surrounding canopy or ground, as it effectively accounts for the morphological differences between such elements. Subsequently, we evaluate the horizontal deviation between the fitted slice center and the predicted tree position. This comparison helps mitigate the effect of outliers or uneven ground, which could otherwise lead to the inclusion of trunk and ground points in the same slice, particularly in areas with sloped terrain. By establishing a tolerance for the covariance and center distance, we account for the natural variability in trunk shape and size, such as differences in diameter and the presence of non-straight or skewed trunks. Upon identifying the trunk successfully, the method advances upward until reaching a slice where the Gaussian model no longer fits the data points. This signifies the transition from trunk to canopy and, thus, the termination of the trunk detection process.

Lastly, a *post-processing step* is designed to further extract and refine the ground points and tree centers throughout the entire model. We begin within each segmented tree area by removing the previously identified trunk points and any points above the trunk, which are classified as canopy. Following this pruning, DBSCAN is utilized to filter out outliers from the ground points. Afterwards, we fit a Gaussian distribution to all the points belonging to the trunk in the X and Y directions. The center of this distribution is determined to be the central position of the tree, and the minimum Z-value among the trunk points is taken as the Z-coordinate for the tree center. This procedure effectively transforms the detailed trunk point data into a singular, representative tree center point. In the final step, ground points from all

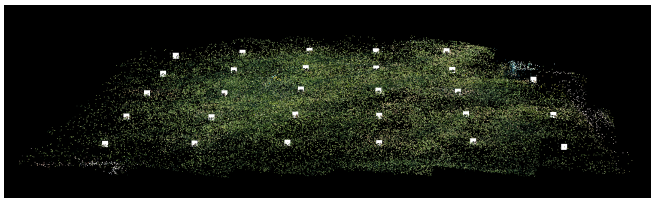


Fig. 5: Example of extracted ground points and trunk centers for a single model. (white points represent the trunk centers)

segmented tree areas within the model are aggregated and an outlier removal process is applied once more, this time across the entire model, to ensure the extraction of a clean, accurate ground representation. Fig. 5 illustrates the feature extraction results from a single model.

D. Alignment Stage

The alignment stage is structured as a three-step process designed to ensure the precise registration of our models.

1) Initial Alignment: Given the potentially low accuracy of GPS data, particularly in the Z direction, GPS-aligned point clouds reconstructed from different image sequences can exhibit noticeable discrepancies and shifts relative to each other. To mitigate these issues, we initially perform an ICP alignment of the ground points extracted from the query model with those from the reference model. This preparatory step is crucial for ensuring that the two models are coarsely aligned at the ground level, thereby setting the stage for the subsequent, more intricate alignment processes. After the initial alignment, the trees in the query model become correctly oriented along the Z direction. Consequently, we reapply the segmentation with the previously extracted tree centers and repeat the feature extraction steps to further extract cleaner ground points and more accurate tree centers.

2) Loop Alignment: The second phase of the alignment stage involves a looped process that alternates between ICP-based alignment of ground points and tree centers to incrementally refine the registration of the two models. The proposed strategy, which we execute for five iterations in total, leverages the complementary strengths of both ground and tree center features: ground points ensure a broad, coarse alignment, while tree centers allow for detailed and precise adjustments.

3) Batch Alignment: The final refinement phase addresses the inherent risk of accumulated drift within the query submodel, especially when it results from a long image sequence. This drift can introduce local alignment errors, which are manifested when the distance vectors between corresponding tree centers from the query and the reference models appear similar locally but diverge significantly across different areas. To mitigate this issue, we first cluster the trees within the query model into several groups or 'batches' based on their positions and their distance vectors to the closest tree in the reference model. This is achieved by performing a Delaunay Triangulation, interconnecting all tree centers to form a network, followed by the assessment of similarity scores across all edge pairs within this network. The calculation of this similarity score involves two components: the $L2$ magnitude difference of the edges ($\Delta\psi$) and their directional difference ($\Delta\theta$).

$$\text{Similarity score} = \exp(-\Delta\psi) \times \left(1 - \frac{\Delta\theta}{\pi}\right)$$

Edges exhibiting a similarity score below a predefined threshold are cut, enabling us to isolate clusters of neighboring tree centers. Then, the centroid of each cluster serves as an initial seed for the k-means clustering algorithm. This process results in well-defined groups of tree centers, each characterized by similar error profiles and spatial proximity. Finally, each of these groups or batches gets aligned individually to the reference model based on the positions of the tree centers.

IV. Experiments

To evaluate the effectiveness of our registration method, we test it on a variety of real-world sequences collected at two different locations, namely an apple farm and a park, with areas of $80\,\mathrm{m}$ by $55\,\mathrm{m}$ and $300\,\mathrm{m}$ by $40\,\mathrm{m}$, respectively. The images have 4000x3000 pixels. Geo-referenced ground-truth camera poses for each image sequence are obtained using the Pix4D 3D reconstruction software and GCPs. We conduct three sets of experiments using sparse point clouds reconstructed from images via the Structure from Motion (SfM) method in COLMAP, and compare our approach against several classical and state-of-the-art methods for point cloud registration: ICP [3], FGR [24], Fast ICP (FICP) [6], Robust ICP (RICP) [6], and TEASER++ [23] with FPFH [33] features. The alignment quality of different query models with respect to the reference model is measured using two key metrics:

- *Camera Poses Error*: For evaluation purposes, the camera poses of the COLMAP-reconstructed reference model are initially aligned with the ground truth. Then, we register each query model to the reference model and directly measure the discrepancy between the camera poses of the aligned query model and their corresponding ground-truth poses. Specifically, we compute the Root Mean Square Error (RMSE) for translation errors and the Mean Rotation Error (MRE) for rotation errors.
- *Tree Centers Error*: We also aim at comparing the tree center locations extracted from the query and the reference models using the approach described in Sec. III-C. To this end, we calculate the Euclidean distance between corresponding tree center pairs.

The next sub-sections present the setup and the results of each experiment, while a discussion of our findings and additional ablation studies are provided in Sec. V.

A. Different Seasons

For the first experiment, we use two image sequences collected from an apple farm in Kastelholf, Switzerland, across different seasons, i.e. winter and summer, as shown in Fig. 6. These sequences consist of 710 and 422 images, respectively. This selection provides a challenging testbed for our method, given the significant changes in natural features

(a) Winter model. (b) Summer model.

Fig. 6: Models to be aligned in the Different Seasons experiment. The winter model is used as reference, and the summer model is used as query.

	Ours	ICP	FGR	FICP	RICP	Teaser++
Translation [m]	**0.44**	1.77	4.66	5.88	5.65	22.58
Rotation [°]	**0.74**	2.48	0.77	6.82	7.50	19.62

TABLE I: Comparison of camera poses error for different algorithms on the Different Seasons experiment.

	Ours	ICP	FGR	FICP	RICP	Teaser++
3D Error [m]	**0.284**	1.875	4.770	5.386	4.951	21.766

TABLE II: Comparison of median tree centers error on the Different Seasons experiment.

	Ours	ICP	FGR	FICP	RICP	Teaser++
East view						
Translation [m]	0.22	0.69	26.66	1.51	**0.18**	60.55
Rotation [°]	0.30	0.97	2.21	1.92	**0.24**	10.73
North view						
Translation [m]	**0.26**	0.88	6.49	4.62	0.83	46.28
Rotation [°]	**0.18**	0.39	1.33	10.60	0.28	165.44

TABLE III: Comparison of camera poses error for different algorithms on the Perpendicular Viewpoints experiment.

	Ours	ICP	FGR	FICP	RICP	Teaser++
3D Error (East) [m]	**0.118**	0.758	26.796	1.436	0.238	61.100
3D Error (North) [m]	**0.196**	0.840	6.533	4.938	0.725	35.996

TABLE IV: Comparison of median tree centers error on the Perpendicular Viewpoints experiment.

and lighting conditions between seasons. Table I compares the translation and rotation errors of camera poses after registering the two sequences with all the evaluated methods, while the tree centers errors are reported in Table II.

B. Perpendicular Viewpoints

Images for the second experiment are sourced from the same apple farm in Kastelholf, Switzerland, as detailed in the Different Seasons case. Here, we utilize the winter model as the reference model, while query models are reconstructed from images captured in summer from the east and north viewpoints, comprising 156 and 260 images, respectively. This setup allows for an analysis of alignment quality in the context of temporal changes and partial observations. Table III compares the translation and rotation errors of cameras across all methods, and Table IV presents a comparison of the tree centers error.

C. Opposite Viewpoints

Images in the third experiment are gathered from Andreas Park, Switzerland. In particular, we use two sequences spanning various views from east to west and west to east in a zigzag pattern, i.e. 180 degrees of viewpoint change, with 497 and 687 images, respectively. The goal here is to evaluate the method's robustness to challenges arising from the absence of identical features and large-scale environments.

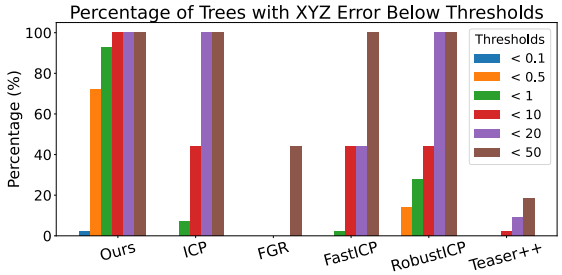


Fig. 7: Comparison of tree centers error for different thresholds.

The reconstructed model from the west-to-east sequence exhibits SfM errors, resulting in disjointed models. Through our pipeline, the image sequence can be automatically divided into two sub-sequences, leading to separate reconstructions corresponding to Submodel 1 and Submodel 2. Consequently, we use the east-to-west model as the reference model and test four cases of query models: 1) the no split COLMAP model, i.e., the COLMAP model using the full sequence, 2) the integrated result of Submodel 1 and Submodel 2, 3) Submodel 1, and 4) Submodel 2.

Table V shows the translation and rotation errors of cameras across all methods for these four cases, while Table VI presents the error in 3D tree centers. To enhance the clarity of the results, we further focus on the integrated model that results from merging Submodel 1 and Submodel 2 and assess the proportion of tree center errors that fall below a set of predefined threshold distances, as depicted in Fig. 7.

V. Discussion

A. Methods Comparison

Compared with other ICP-based methods, including ICP, FastICP, and RobustICP, our method demonstrates consistently better or comparable results. The superiority evidenced by the camera pose errors detailed in Table I, Table V, as well as the tree centers error in Table II, Table IV, Table VI, highlights the effectiveness of aligning carefully extracted features compared to utilizing all points, thereby mitigating the effects of noise and data sparsity. An exception is noted in the Perpendicular Viewpoints experiment (Table III), where RICP outperforms our method on the East view model. However, our method is closely competitive, lagging by only $4\,\mathrm{cm}$ in translation and $0.06°$ in rotation. Despite the lower accuracy in camera pose estimation, our method still achieves the best result for tree center errors, as shown in Table IV, validating the effectiveness of batch alignment.

Furthermore, our approach presents the lowest errors in all experiments on camera poses (Table I, Table III, Table V) and tree centers (Table II, Table IV, Table VI) relative to the state-of-the-art methods FGR and TEASER++ with FPFH features. These two approaches struggle to accurately identify features in natural environment, leading to alignment

	Translation Error [m]						Rotation Error [°]					
	Ours	**ICP**	**FGR**	**FICP**	**RICP**	**Teaser++**	**Ours**	**ICP**	**FGR**	**FICP**	**RICP**	**Teaser++**
All - no split Colmap	**0.40**	17.49	84.89	27.97	21.75	151.13	**0.54**	3.31	0.99	3.36	4.10	148.77
All - Submodel 1 and Submodel 2	**0.40**	15.83	118.37	23.75	14.80	200.77	**0.54**	3.64	1.29	3.92	2.35	158.65
Submodel 1	**0.60**	3.85	42.40	4.09	1.15	207.24	**0.53**	3.56	2.80	3.88	0.75	17.42
Submodel 2	**0.30**	17.06	119.16	25.15	15.6	146.93	**0.54**	3.66	1.16	3.93	2.42	160.55

TABLE V: Comparison of camera poses error for different algorithms on the Opposite Viewpoints experiment.

	3D Error [m]					
	Ours	**ICP**	**FGR**	**FICP**	**RICP**	**Teaser++**
All - no split Colmap	**0.318**	17.506	84.691	27.954	21.790	125.145
All - Submodel1 and Submodel2	**0.318**	14.931	107.922	22.911	14.058	203.165
Submodel 1	**0.261**	1.790	41.135	1.866	0.716	205.836
Submodel 2	**0.456**	16.702	108.993	24.698	15.430	97.842

TABLE VI: Comparison of median tree centers error on the Opposite Viewpoints experiment.

	Colmap initial	**Initial aligned**	**Loop aligned**	**Batch aligned**
East View				
Horizontal Error (x-y plane) [m]	2.460	1.411	0.095	0.086
Vertical Error (z direction) [m]	90.390	0.220	0.090	0.080
3D Error [m]	90.430	1.412	0.180	**0.118**
North View				
Horizontal Error (x-y plane) [m]	3.274	2.993	0.136	0.121
Vertical Error (z direction) [m]	99.395	0.140	0.120	0.130
3D Error [m]	99.429	2.996	0.200	**0.196**

TABLE VII: Comparison of median tree centers error for each stage of Perpendicular Viewpoints case.

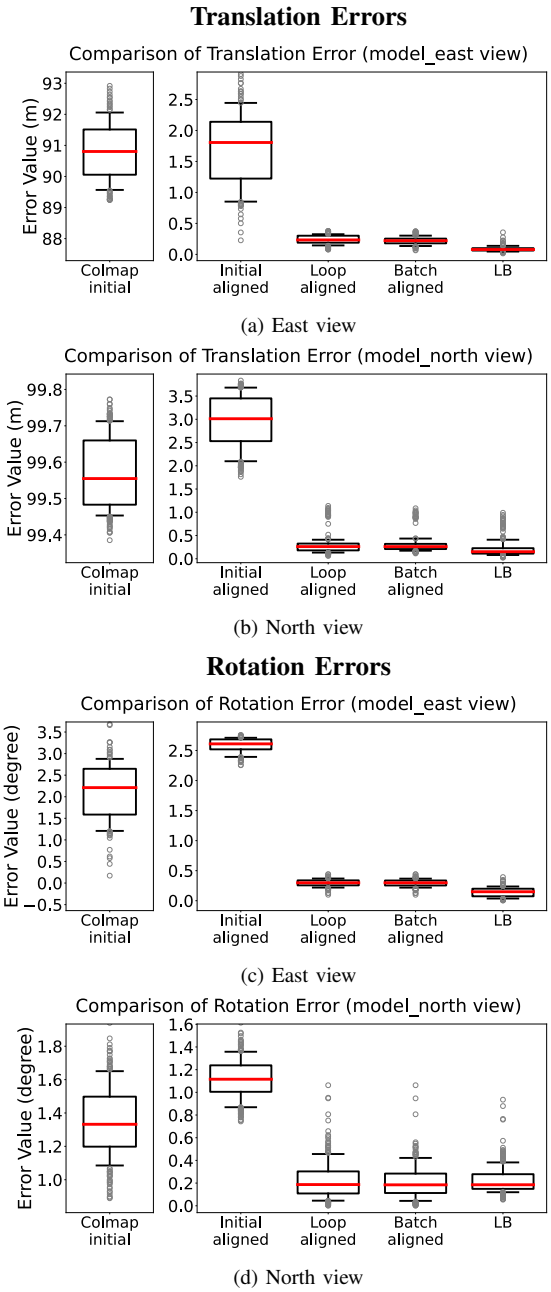
failure. In contrast, our method provides a robust feature extraction process that facilitates subsequent alignment.

B. Ablation Study

To evaluate the effectiveness of each step in our pipeline, we utilized the Perpendicular Viewpoints experiment. We compared the error in tree centers at various stages: the initial COLMAP model, after initial alignment, after loop alignment, and after batch alignment. The results, presented in Table VII and Fig. 8, demonstrate that:

1) The effectiveness of the initial alignment in the vertical direction is evident, especially when comparing the vertical error of tree centers from the COLMAP initial models to the initial aligned models.
2) The errors in both camera poses and tree centers after loop alignment demonstrate the three-dimensional effectiveness of loop alignment.
3) There is a further improvement in alignment with batch alignment, indicating its efficacy in enhancing model accuracy.

In order to differentiate alignment imperfections from COLMAP reconstruction errors, we also measured the camera pose errors obtained when directly aligning the query model to the ground truth. This can provide an indication of the lower bound for the alignment error, shown as 'LB' in Fig. 8. For the East view model, the reconstruction error amounts to $8\,\mathrm{cm}$ in translation and $0.15°$ in rotation, while



(a) East view

(b) North view

(c) East view

(d) North view

Fig. 8: Comparison of camera poses error for each stage of Perpendicular Viewpoints case.

for the North view model, it is $15\,\mathrm{cm}$ and $0.19°$, respectively. Due to the closeness of these values to our final alignment errors, it appears that the primary source of inaccuracies is the original reconstruction itself, from COLMAP.

VI. Conclusion

In this paper, we develop a pipeline for aligning multiple SfM models of natural environments with planted trees. Key to our approach is the employment of a learning-based perspective to bird's-eye view image conversion for estimating the positions of trees, a feature extraction strategy that targets the clean segmentation of ground and trunk points, and an iterative process that leverages these features for precise alignment. The proposed method is evaluated on challenging experiments, exhibiting notable robustness to high levels of noise in the input point clouds and significantly outperforming well-established point-cloud registration methods in terms of alignment accuracy.

Future directions include investigating strategies to increase the accuracy and repeatability of the tree-center extraction process. In addition, a final re-optimization step through bundle adjustment to further refine the estimated camera poses will also be considered.

Acknowledgments

We thank Sarah-Lia Wotke for her contribution to the preliminary version of this work, Junyuan Cui for the ground-truth generation for the BEV network, and Peter Fröhlich/AgriCircle for access to the orchard.

References

[1] X. Liang, A. Kukko, I. Balenović, N. Saarinen, S. Junttila, V. Kankare, and ..., "Close-range remote sensing of forests: The state of the art, challenges, and opportunities for systems and data acquisitions," *IEEE Geoscience and Remote Sensing Magazine*, vol. 10, no. 3, 2022.

[2] L. Bartolomei, L. Teixeira, and M. Chli, "Fast multi-uav decentralized exploration of forests," in *IEEE Robotics and Automation Letters*, 2023.

[3] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. Spie, 1992, pp. 586–606.

[4] A. L. Pavlov, G. W. Ovchinnikov, D. Y. Derbyshev, D. Tsetserukou, and I. V. Oseledets, "Aa-icp: Iterative closest point with anderson acceleration," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3407–3412.

[5] S. Rusinkiewicz, "A symmetric objective function for icp," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–7, 2019.

[6] J. Zhang, Y. Yao, and B. Deng, "Fast and robust iterative closest point," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3450–3466, 2021.

[7] Y. Wang and J. M. Solomon, "Deep closest point: Learning representations for point cloud registration," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3523–3532.

[8] Z. Dang, L. Wang, Y. Guo, and M. Salzmann, "Match normalization: Learning-based point cloud registration for 6d object pose estimation in the real world," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–15, 2024.

[9] W. Yuan, B. Eckart, K. Kim, V. Jampani, D. Fox, and J. Kautz, "Deepgmr: Learning latent gaussian mixture models for registration," in *European Conference on Computer Vision(ECCV)*, 2020.

[10] W. Dai, B. Yang, X. Liang, Z. Dong, R. Huang, Y. Wang, J. Pyörälä, and A. Kukko, "Fast registration of forest terrestrial laser scans using key points detected from crowns and stems," *International Journal of Digital Earth*, vol. 13, no. 12, pp. 1585–1603, 2020.

[11] X. Wang, Z. Yang, X. Cheng, J. Stoter, W. Xu, Z. Wu, and L. Nan, "Globalmatch: Registration of forest terrestrial point clouds by global matching of relative stem positions," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 197, p. 71–86, Mar. 2023.

[12] X. Ge, Q. Zhu, L. Huang, S. Li, and S. Li, "Global registration of multiview unordered forest point clouds guided by common subgraphs," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2021.

[13] S. Rumley, A. Thoma, P. Beardsley, L. Teixeira, and M. Chli, "From perspective view to bird's eye view in agricultural environments," in *40th IEEE Conference on Robotics and Automation Workshops(ICRAW 2023)*, 2023.

[14] X. Huang, G. Mei, J. Zhang, and R. Abbas, "A comprehensive survey on point cloud registration," *arXiv preprint arXiv:2103.02690*, 2021.

[15] Y. He, J. Yang, X. Hou, S. Pang, and J. Chen, "Icp registration with dca descriptor for 3d point clouds," *Optics express*, vol. 29, no. 13, pp. 20 423–20 439, 2021.

[16] L. He, S. Wang, Q. Hu, Q. Cai, M. Li, Y. Bai, K. Wu, and B. Xiang, "Gfoicp: Geometric feature optimized iterative closest point for 3-d point cloud registration," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–17, 2023.

[17] O. Hirose, "A bayesian formulation of coherent point drift," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 7, pp. 2269–2286, 2021.

[18] W. Gao and R. Tedrake, "Filterreg: Robust and efficient probabilistic point-set registration using gaussian filter and twist parameterization," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[19] B. Eckart, K. Kim, and J. Kautz, "Hgmr: Hierarchical gaussian mixtures for adaptive 3d registration," in *proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 705–721.

[20] H. Hong and B. H. Lee, "Probabilistic normal distributions transform representation for accurate 3d point cloud registration," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 3333–3338.

[21] J. Huang, B. Tao, and F. Zeng, "Point cloud registration algorithm based on icp algorithm and 3d-ndt algorithm," *International Journal of Wireless and Mobile Computing*, vol. 22, no. 2, pp. 125–130, 2022.

[22] H. M. Le, T.-T. Do, T. Hoang, and N.-M. Cheung, "Sdrsac: Semidefinite-based randomized approach for robust point cloud registration without correspondences," in *IEEE/CVF conference on computer vision and pattern recognition*, 2019.

[23] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 314–333, 2020.

[24] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer, 2016, pp. 766–782.

[25] C. Gao, F. Daxinger, L. Roth, F. Maffra, P. Beardsley, M. Chli, and L. Teixeira, "Aerial image-based inter-day registration for precision agriculture," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[26] Z. J. Yew and G. H. Lee, "Regtr: End-to-end point cloud correspondences with transformers," in *IEEE/CVF conference on computer vision and pattern recognition*, 2022.

[27] W. Lu, G. Wan, Y. Zhou, X. Fu, P. Yuan, and S. Song, "Deepvcp: An end-to-end deep neural network for point cloud registration," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 12–21.

[28] F. Ghorbani, Y.-C. Chen, M. Hollaus, and N. Pfeifer, "A robust and automatic algorithm for tls-als point cloud registration in forest environments based on tree locations," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2024.

[29] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[30] P. Moulon, P. Monasse, R. Perrot, and R. Marlet, "OpenMVG: Open multiple view geometry," in *International Workshop on Reproducible Research in Pattern Recognition*. Springer, 2016, pp. 60–74.

[31] S.-Y. Yang, W.-C. Lee, Y.-C. F. Wang, and Y.-K. Lai, "Projecting your view attentively: Monocular road scene layout estimation via cross-view semantic alignment," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[32] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[33] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3212–3217.