UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours
Final Paper 2021

Title: Evaluation of Point Cloud Registration Methods on Multitemporal Orthomosaic Alignment

Author: Damian Wilson WLSDAM001

Project Abbreviation: TreeReg

Supervisor(s): Professors Patrick Marais and Deshen Moodley

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	
Theoretical Analysis	*0*	*25*	
Experiment Design and Execution	*0*	*20*	10
System Development and Implementation	*0*	*20*	15
Results, Findings and Conclusions	*10*	*20*	20
Aim Formulation and Background Work	*10*	*15*	15
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	
Total marks		**80**	

Evaluation of Point Cloud Registration Methods on Multitemporal Orthomosaic Alignment

TREEREG CS Honours Project 2021

Damian Wilson
Department of Computer Science
University of Cape Town
Cape Town, South Africa

ABSTRACT

This project evaluated the applicability of two different Point Cloud Registration (PCR) methods to the problem of multitemporal orchard orthomosaic registration. Each orthomosaic has an associated set of feature detections, in the form of bounding polygons over individual tree canopies. The goal of this project was to determine whether PCR could use these detections to accurately register two orthomosaic maps of the same orchard, taken at different times. PCR refers to the alignment of two separate point sets, where the source point cloud is transformed to overlay the reference cloud. This problem is transformed into a PCR problem through using the centroids of each images' tree detections as the input source clouds to a PCR method. Two different rigid PCR methods were implemented for this purpose. The first is Iterative Closest Point (ICP), this algorithm turns the problem of PCR into an optimization problem in which rigid registration parameters are estimated with the goal of reducing the distance between matched point pairs iteratively. The second is a newer, probabilistic approach to PCR, called Coherent Point Drift (CPD), where the source point cloud is modelled as a Guassian Mixture Model (GMM) and the reference point cloud as observations from that GMM. Registration parameters are estimated that maximize the probability that the data points in the reference cloud belong to the GMM. Both methods had their resulting registration accuracy levels tested over a wide range of rigid deformations on the input data. It was shown that CPD outperformed ICP in registration accuracy in all cases and produced results to an acceptable degree of accuracy, meeting the aim of this project in the case of rigid deformation of the input data. Additionally, CPD was shown to be more robust to unforeseen deformations in the input dataset, such as the addition of outlier/noise and the removal of low confidence detections. ICP was not capable of providing an acceptable level of registration accuracy.

KEYWORDS

Image Registration, Point Cloud Registration, Iterative Closest point, Coherent Point Drift

1 Introduction

Precision Agriculture (PA) is a modern farming management concept based on observing, measuring and responding to inter and intra-field variability in crops. The goal of which is to maximise

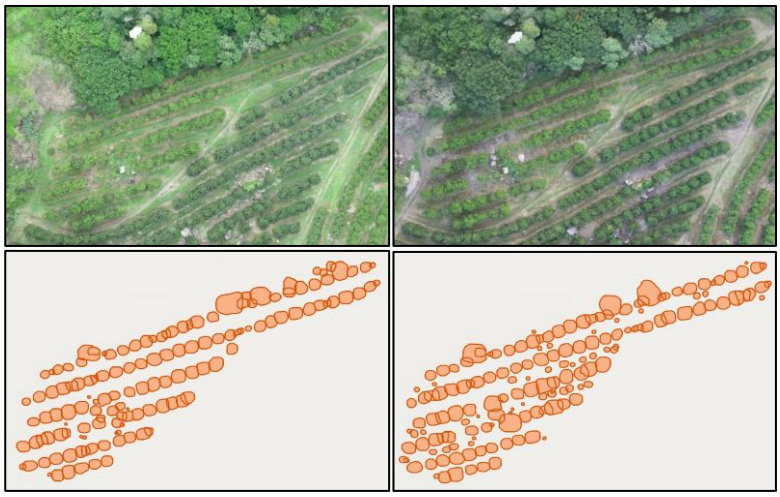

Figure 1: Multi-temporal orthomosaic maps of an orchard (top) and their corresponding feature detections (bottom).

crop yield while minimising resource expenditure [1], [2], [3]. The advancement and use of Unmanned Aerial Vehicles (UAV) are a driving force behind the viability of current PA practices. This is due to their ability to take high resolution images of farmland on demand in a cost-effective manner [3]. These images are subsequently processed, aligned and stitched together to form orthomosaic maps over a large stretch of farmland. An orthomosaic is a high-resolution geometrically corrected aerial image, composed of many individual still images being stitched together [4]. These orthomosaics are key inputs to agricultural analysis applications for PA.

Aerobotics [6], a stakeholder in this project, is a South African company the PA sector. They provide agricultural trend analysis for farmers using multi-temporal orthomnosaic maps of orchards as input. A core functionality of their product is the ability to compare changes in tree performance (on an individual tree-level) over time. For this to be possible, it is critical that the input orthomosaics are correctly registered. Image registration is the process of correctly overlaying two or more images of the same scene [5]. The images could be taken from different angles, sensors or times. These orthomosaic maps are tagged with GPS data by the drone when the pictures are taken. Due to the inherent error in this tagging, the maps are slightly misalingned between dates. They currently use a semi-automated approach to register these maps, where successive surveys (orthomosaic maps) are registered with previously processed datasets using a human-in-the-loop approach. This reliance on manual intervention in the registration process creates limitations in the scalability of their orchard trend analysis system. The automation of this registration process is the crux of this project.

The detection and matching of salient features in the input images is a common approach to image registration [5]. Aerobotics have developed a robust feature extractor for this purpose, a tree instance segmentation model. This produces a bounding polygon around individual tree canopies on the map, with an associated confidence level regarding the accuracy of the detection (See figure 1). The main aim of this project is to use these tree detections as inputs to a feature-based method that automatically registers the input orthomosaics. The method must correctly overlay the newer othomosaic on to the older one, enabling per-tree agricultural trend analysis. A key requirement of traditional feature-based registration methods is the use of invariant features as inputs to be matched between images. These correspondences are then used to estimate image transformation parameters to overlay the source image on the reference [5]. The use of tree detections as inputs to traditional feature-based registration methods breaks the invariance requirement as trees are inherently temporally unstable. They are subject to grow, wither or die over time and as such their corresponding detections change. Therefore, alternative registration methods using these features need to be explored. This project aims to evaluate the use of Point Cloud Registration (PCR) methods for this problem.

PCR is the process of estimating the transformation parameters that align two input point clouds [7]. The problem of tree-feature based registration can be converted to a PCR problem using the centroids of individual polygon detections as points, making up each of the input point clouds. The applicability of two rigid PCR methods to this problem will be tested. The first is Iterative Closest Point (ICP). ICP is a classic and simple approach to PCR [8]. ICP transforms the problem of image alignment into an optimization problem. It involves the estimation of transformation parameters to minimize an error metric (i.e., distance between corresponding points) in iterations. Points between the input point clouds are matched based on closest proximity. However, traditional ICP is very sensitive to the presence of outliers in the dataset (points with no correspondence in the opposite point cloud) and tends to fall into local minima, preventing accurate registrations of complex data [9]. Therefore, a second method will also be considered as a possible solution. The Coherent Point Drift (CPD) algorithm is a more recent approach to PCR, presented in 2010 by Andriy Myronenko and Xubo Song [10]. Unlike ICP, CPD makes use of a probabilistic approach to estimate registration transformation parameters. Points in the source cloud are modelled as centroids of a Guassian Mixture Model (GMM) and points in the reference cloud as observations of the GMM. Transformation parameters that maximize the posterior probability that the GMM observations belong to the GMM are estimated iteratively, producing an accurate registration. This probabilistic approach has been shown to mitigate the adverse effect of outliers on registration accuracy. Both these methods will be explained. They will be tested using real feature detections provided by Aerobotics. Factors such as registration accuracy, computation time and robustness will be recorded and compared. From this, conclusions will be drawn over their applicability to the problem domain.

2 Related Work

In this section, registration techniques related to the two algorithms being evaluated will be reviewed. As this project requires the registration of images using detected features (i.e., tree polygons), relevant feature-based registration methods will be briefly reviewed. Followed by a review on other PCR methods, related to the chosen two for this project.

Feature-based registration methods: Feature based registration methods make use of the extraction, matching and subsequent aligning of salient features between the two input images [19]. The salient features that would be used in this case are the tree instance polygons (Figure 1). These methods match features based on the correspondence of their descriptions (i.e., the level of similarity between the tree polygons in this case) [5], [20]. Established and widely used feature-based registration methods include the Scale Invariant Feature Transform method (SIFT) [20], [21], and it's newer and faster variation, Speeded Up Robust Feature method (SURF) [20], [22]. A key requirement for these methods to be successful is that the feature descriptors remain invariant over time [5]. This is not possible in this case as the shape of the tree canopies being detected are inherently temporally unstable and prone to change over time. Therefore, other approaches to registration must be considered in this project.

Point Cloud Registration methods: The goal of PCR methods is to align two separate point sets so that the source point set is transformed to accurately overlay the reference point set [7]. This is done through finding corresponding points in the input point clouds and estimating transformation parameters to accurately overlay these matched points. PCR methods are generally divided in to 2 categories, rigid point set registration algorithms and non-rigid registration algorithms [7], [10], [16]. Rigid PCR algorithms estimate rigid transformation parameters to register the point sets, these being transformations that preserve the distance between all point pairs in the cloud (i.e., rotation and translation). Non-rigid PCR methods don't have this restriction. This project will only test rigid PCR algorithms on the provided input data. ICP, one of the algorithms being implemented, is the most popular rigid PCR method due to its simplicity and low computational complexity [10]. The ICP method being implemented will be further explained below. Traditional ICP is shown to fail if the input source clouds are not adequately close or there are a significant proportion of outliers in the dataset [9], [10]. Therefore, many probabilistic rigid PCR methods have been developed to overcome these ICP limitations [23], [24]. These methods assign a soft correspondence between all combinations of possible matched pairs of points between the input clouds, according to some probability. As opposed to the hard correspondences assigned based on closest distance in ICP. An example of this type of probabilistic method is Robust Point Matching (RPM) [25] and it's later variants [26]. Another class of probabilistic PCR methods include methods that model the source cloud as centroids of a Guassian Mixture Model (GMM), and the reference cloud as observations from that GMM, of which CPD [10] is one. Several rigid PCR methods, including [27], [28], [29], formulate point cloud registration as a maximum likelihood estimation problem. The GMM centroids are transformed with estimated rigid parameters to maximize the likelihood that the datapoints in the reference point cloud are observations of that GMM. These GMM-based algorithms are generalized as Expectation-Maximization (EM) algorithms as they follow the following two basic steps iteratively: The Expectation step consists of determining the probability that the points in the reference cloud are observations of the GMM. The subsequent Maximization step estimates rigid transformation parameters to maximize this probability [29]. These steps are repeated iteratively until convergence in the estimation of registration parameters. These probabilistic methods perform better than ICP, especially in the presence of outliers. CPD was selected for this project, over these other probabilistic methods, due to it's status as a state-of-the-art method, with high rates of accuracy, for both rigid and non-rigid registration [17].

3 Methodology

This orthomosaic registration problem will be transformed in to a PCR problem, with two different methods being implemented as possible solutions. The results produced by each method will be compared and their applicability to the problem evaluated.

3.1 ICP Algorithm Overview

The traditional ICP algorithm involves estimating *rigid* transformation parameters (rotation and translation) to transform a source point cloud on to a reference point cloud iteratively. With the goal of each iteration being to minimize an error metric, being the Euclidean distance between closest point pairs of the two-point clouds.

The ICP algorithm follows the following basic steps [9]:

1. For two sets of points M and S, corresponding to the reference and source point clouds, calculate the nearest point in set M for every point in set S.
2. Compute the rotation matrix R and translation vector t using the least squared method for distance minimization and Singular Value Decomposition (SVD). Equations 1 to 8.
3. Transform set Y by the R and t.
4. Compute the error E between the transformed set and the reference point sets. Equation 9.
5. Repeat in iterations until convergence in error reduction.

Once the corresponding points in step one are known, the transformations in point 2 are calculated using SVD. SVD computation requires centroid alignment and is calculated as follows [9]:

$$C_S = \frac{1}{N}\sum_{i=1}^{N} s_i \tag{1}$$

$$C_M = \frac{1}{N}\sum_{i=1}^{N} m_i \tag{2}$$

Next, we align the points in M and S using their centroid:

$$\mathcal{M}' = \{m_i' = m_i - C_M\}_{1,\dots,N} \tag{3}$$

$$\mathcal{S}' = \{s_i' = s_i - C_S\}_{1,\dots,N} \tag{4}$$

Using these values, we compute the covariance matrix H:

$$H = \mathcal{M}'\mathcal{S}'^{T} \tag{5}$$

Then compute the SVD as:

$$H = U\Lambda V^T \tag{6}$$

The R and t are calculated as follows:

$$R = VU^T \tag{7}$$

$$\mathbf{t} = C_S - RC_M \tag{8}$$

After R and t have been calculated in the iteration, the error of the transformation must be determined:

$$E(R,t) = \sum_{i=1}^{N_M} \sum_{i=1}^{N_S} w_{ij} \|m_i - (Rs_j + t)\|^2 \quad (9)$$

Where w is the weight of corresponding points. If point m_i is the closest point to the transformed point it's set to 1, else it is 0. This ensures only closest point pairs are used to determine error.

This process repeats until convergence in error reduction or transformation parameter estimation.

K-d Tree for Nearest Neighbor Search:

The most computationally intensive operation in an iteration of the ICP algorithm is the nearest neighbor search, where the nearest point in the reference point cloud for every point in the source cloud is found. A brute force search would be an $O(N^2)$. This can drastically affect the computation time of an ICP registration. An alternative to this brute force search is to use a data structure called a k-d tree.

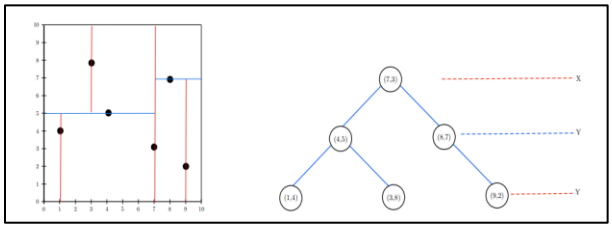

Figure 2: Building a 2-D k-d tree (left) and a binary representation of a k-d tree (right)

A k-d tree is a way of storing points of k-dimensions in a Binary Search Tree. It's a powerful tool for nearest neighbor searches, boasting an O(logN) computational complexity for a nearest point search [12]. The k-d tree will be used to maximise the computational efficiency of the implemented ICP algorithm. It has been shown to decrease the computation time of an ICP algorithm up to 10 times [30].

3.2 Coherent Point Drift (CPD) Algorithm

CPD is a newer PCR algorithm. It was introduced as a way to accurately register point clouds despite the presence non-rigid deformations, noise and outliers in the dataset. Traditional PCR methods, like ICP, are known to fail when these adverse conditions are present in the input data [10]. CPD counters this through using a probabilistic approach to registration. Unlike ICP, where registration is modelled as a distance error minimization problem, CPD considers the alignment of point clouds as a probability density estimation problem. The source point cloud is modelled as a Guassian Mixture Model (GMM). A GMM is defined as a weighted sum of individual Guassian components [11]. They are used to represent a fixed number of normally distributed subpopulations within an overall population. Each point in the source point cloud is modelled as a centroid of an individual normally distributed Guassian component within the GMM. The reference point cloud points are modelled as observations from that GMM. Transformation parameters are estimated to maximize the likelihood that the data points (reference) belong to the GMM iteratively. The GMM centroids move coherently in accordance to these transformation parameters to preserve the topological structure of the source cloud. Unlike ICP, where point correspondence is definite and based on Euclidean distance, CPD provides a soft correspondence of points based on probabilities instead of distance. Each point in the reference cloud has a probability that it corresponds to each individual point in the source cloud which are modelled as individual normal distributions in the GMM. This probabilistic approach to point correspondence lessens the potential impact of outliers and noise on overall registration accuracy. Registration transformation parameters are calculated to maximize the overall probability that the reference data points belong to the GMM, aligning them. Whereas in ICP the transformation parameters are calculated based on potentially false point correspondences due to outliers [9]. Another key feature of the CPD algorithm is it's ability to register point sets with non-rigid deformations, unlike ICP. CPD registers point sets using an Expectation-Maximization algorithm (EM) in which the probability that the reference data points belong to the source GMM is calculated, and then transformation parameters to maximize this probability are calculated in iterations until convergence. The steps of the CPD algorithm will be provided below, in the case of rigid transformations in order to be compared with ICP.

CPD steps for Rigid Point Set Registration [10]:

Rigid point set registration algorithm:

- Initialization: $\mathbf{R} = \mathbf{I}, \mathbf{t} = 0, s = 1, 0 \leq w \leq 1$

  $\sigma^2 = \frac{1}{DNM} \sum_{n=1}^{N} \sum_{m=1}^{M} \|\mathbf{x}_n - \mathbf{y}_m\|^2$
- EM optimization, repeat until convergence:
  - E-step: Compute $\mathbf{P}$,

    $p_{mn} = \frac{\exp -\frac{1}{2\sigma^2}\|\mathbf{x}_n - (s\mathbf{R}\mathbf{y}_m + \mathbf{t})\|^2}{\sum_{k=1}^{M} \exp -\frac{1}{2\sigma^2}\|\mathbf{x}_n - (s\mathbf{R}\mathbf{y}_k + \mathbf{t})\|^2 + (2\pi\sigma^2)^{D/2} \frac{w}{1-w} \frac{M}{N}}$
  - M-step: Solve for $\mathbf{R}, s, \mathbf{t}, \sigma^2$:
    - $N_{\mathbf{P}} = \mathbf{1}^T\mathbf{P}\mathbf{1}, \mu_{\mathbf{x}} = \frac{1}{N_{\mathbf{P}}}\mathbf{X}^T\mathbf{P}^T\mathbf{1}, \mu_{\mathbf{y}} = \frac{1}{N_{\mathbf{P}}}\mathbf{Y}^T\mathbf{P}\mathbf{1},$
    - $\hat{\mathbf{X}} = \mathbf{X} - \mathbf{1}\mu_{\mathbf{x}}^T, \ \hat{\mathbf{Y}} = \mathbf{Y} - \mathbf{1}\mu_{\mathbf{y}}^T,$
    - $\mathbf{A} = \hat{\mathbf{X}}^T\mathbf{P}^T\hat{\mathbf{Y}}$, compute SVD of $\mathbf{A} = \mathbf{U}SS\mathbf{V}^T,$
    - $\mathbf{R} = \mathbf{U}\mathbf{C}\mathbf{V}^T$, where $\mathbf{C} = \mathrm{d}(1, .., 1, \det(\mathbf{U}\mathbf{V}^T)),$
    - $s = \frac{\mathrm{tr}(\mathbf{A}^T\mathbf{R})}{\mathrm{tr}(\hat{\mathbf{Y}}^T \mathrm{d}(\mathbf{P}\mathbf{1})\hat{\mathbf{Y}})},$
    - $\mathbf{t} = \mu_{\mathbf{x}} - s\mathbf{R}\mu_{\mathbf{y}},$
    - $\sigma^2 = \frac{1}{N_{\mathbf{P}}D}(\mathrm{tr}(\hat{\mathbf{X}}^T \mathrm{d}(\mathbf{P}^T\mathbf{1})\hat{\mathbf{X}}) - s\,\mathrm{tr}(\mathbf{A}^T\mathbf{R})).$
- The aligned point set is $\mathcal{T}(\mathbf{Y}) = s\mathbf{Y}\mathbf{R}^T + \mathbf{1}\mathbf{t}^T,$
- The probability of correspondence is given by $\mathbf{P}$.

Figure 3: CPD rigid registration algorithm [10]

Notation explained:

- D: Dimension of Point Set (2-dimensional in this case)
- N, M: Number of points in each set.
- X_{NxD} : Reference point set (the data point observations of the GMM)
- Y_{MxD}: Source point set (individual GMM centroids)

- T(Y, θ) - Transformation T applied to Y, where θ is a set of the transformation parameters (translation, scaling and rotation)
- R: Rotation Matrix (D x D)
- t: Translation vector (size D)
- s: Scaling parameter (float)
- I: Identity matrix
- 1: Column vector of 1's
- d(v): Diagonal Matrix formed from vector v

In the initialization step, the initial error metric of alignment between the two clouds is calculated. This is variance of the reference data points (X) belonging to the GMM (created by modelling source points Y as GMM centroids). If the clouds were perfectly aligned, with no outliers, this value would be 0. The goal of subsequent stages of the algorithm are to reduce this metric via Expectation and Maximization (EM) steps. In the Expectation step, the probability of cloud alignment is calculated. The goal of the subsequent Maximization step is to calculate new transformation parameters to maximize this probability of alignment. The GMM centroids are then transformed according to these estimations and a new variance error metric is calculated. This is done in iterations until convergence i.e., the reduction in variance falls below a given threshold. The final CPD aligned point set will be given through transforming the source point cloud with the final estimated transformation parameters.

4 Algorithm Implementation and Evaluation

In order to assess the applicability of each PCR algorithm, they both have to be implemented in a software system and have their registration accuracy evaluated using the input data from Aerobotics.

4.1 Input Data

The input data to this project was provided by Aerobotics. It consists of data pertaining to three separate orchards. Each orchard has data corresponding to two separate *surveys* of that orchard: a specific point in time at which a drone captures imagery of the orchard. Each survey contains the tree feature detections (in the form of bounding polygons) of the orchard at that date, as seen in Figure 1. These features are provided in the form of polygon shapes in geojson format. It's important to note that the feature detections of the 2 different dates provided came pre-registered by Aerobotics' current human-in-the-loop system. As such they already are accurately overlayed, according to a human observer. The aim of this project is to accurately overlay one survey (the later one) over another with PCR. However, PCR requires two point clouds as input, a *reference* cloud and a *source* cloud. The source is the cloud that will be transformed to be overlayed on the reference. In the context of this project, the source cloud will correspond to the survey taken at the later date of the two. The centroids of the input tree segment polygons will be used as the input point clouds. The use of centroids as opposed to full polygons as features for registration is more likely to result in accurate registration. This is due to them having a relatively less significant rate of change over time. The temporally unstable nature of tree canopies makes the use of whole tree polygon correspondence between surveys unwise. Tree shapes can change over time, resulting in false matches that potentially invalidate the registration process.

The data is pre-processed as follows:

1. Centroids of each polygon is calculated.
2. The coordinate of each centroid is projected from a Geographic Coordinate System to a simpler 2D pixel coordinate system to avoid complexity and floating point errors in calculations. This is done by scaling the points up by a factor of 100 000 and translating them closer to the origin with the same transformation.

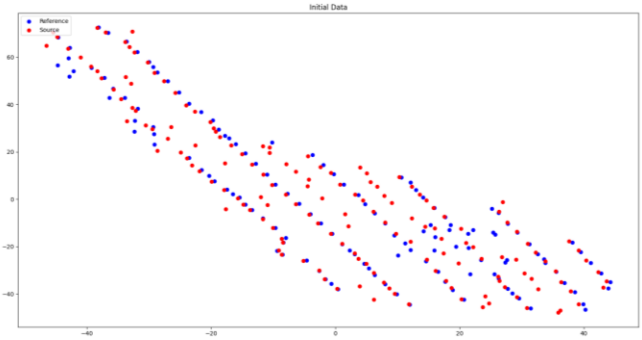

Figure 4: Orchard 1 Input Point Clouds

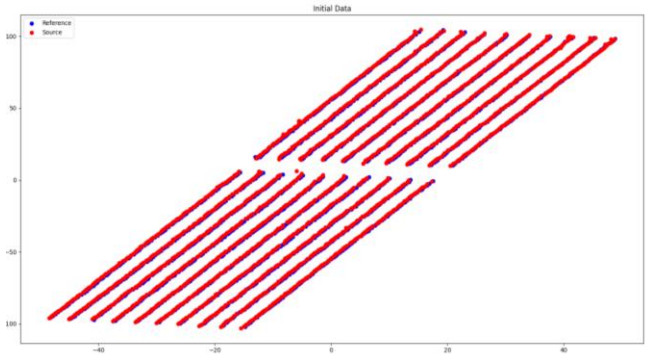

Figure 5: Orchard 2 Input Point Clouds

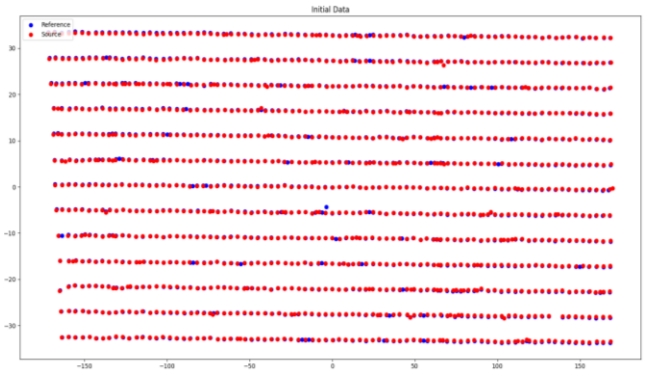

Figure 6: Orchard 3 Input Point Clouds

The result of the preprocessing is shown in Figures 4-6. The detections corresponding to the survey at the earlier date is modelled as the reference point cloud (blue points). The detections of the later survey are modelled as the source point cloud (red points), they overlay the reference points and will be misregistered and subsequently registered via each algorithm during the evaluation process. Orchards 2 and 3 follow a relatively strict row-like planting pattern. Whereas orchard 1 follows a less structured planting pattern, with more random dispersion of tree detections. This contrast in tree planting patterns will test each algorithm's robustness to varying types of orchards as input data.

In addition to a bounding polygon, each detection has a corresponding confidence level. This was produced by Aerobotics' tree instance segmentation model, or feature extractor, and indicates the confidence that the associated polygon is a true detection of a real tree.

Survey	Min	Median	Max	Mean	Std. Dev.	Number of Points
Orchard 1						
1	0.050	0.955	0.998	0.726	0.356	137
2	0.050	0.771	0.998	0.631	0.361	160
Orchard 2						
1	0.051	0.665	0.981	0.659	0.104	1891
2	0.051	0.674	0.949	0.658	0.093	2332
Orchard 3						
1	0.051	0.985	0.999	0.919	0.219	1154
2	0.051	0.987	0.999	0.920	0.216	1152

Table 1: Five Number Summary of Input Data Confidence Levels

As shown in table 1, the input data contains features with a wide range of confidence levels. Orchards 1 and 2 have relatively low average confidence levels, compared to 3. Features with low confidence levels are more likely to be outliers as it's unlikely they correspond to real trees that will be picked up in the other survey. Therefore, orchards 1 and 2 appear to have relatively more outliers than 3. Thus, a registration method with fixed feature correspondences, like ICP, is unlikely to provide an accurate registration in orchards 1 and 2. This is due to the increased likelihood of false point matches with outliers between the clouds.

4.2 Evaluation

Each registration method must be tested on the input data in order to determine their applicability to the problem posed by the project: Can a PCR method accurately register successive orthomosaic surveys using tree feature detections? Each method will be evaluated based on three factors: Registration accuracy, computation time and robustness.

4.2.1 Rigid registration accuracy

The registration accuracy of each method will be recorded after the following rigid deformations are applied to the input source point cloud:

- **Rotation:** The source point cloud will undergo varying degrees of rotational deformations (from 5 to 90 degrees)

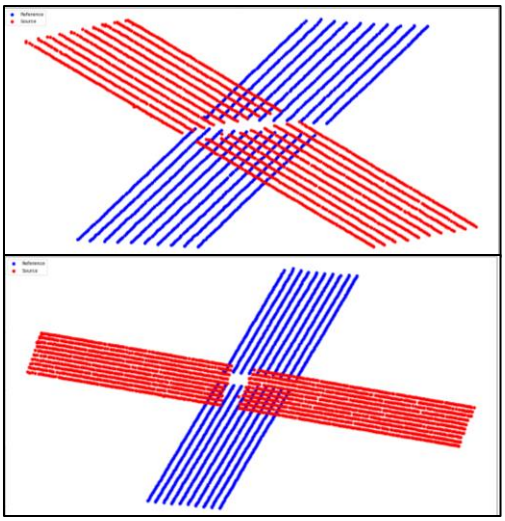

Figure 7: Orchard 2 with 45 (top) and 90 (bottom) degree rotational deformations

- **Translation:** The source point cloud will undergo varying levels of translational deformations. The source cloud will be increasingly translated based on the cloud's respective height and width. The translation vector will

be <percent*width, percent*height>, with percent values ranging from 0.1 to 1.

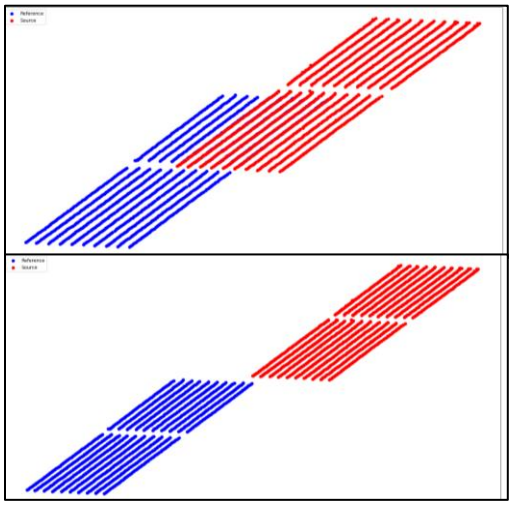

Figure 8: Orchard 2 with 50 (top) and 100 (bottom) percent translational deformations

- **Rotation and Translation:** The source point cloud will undergo a full rigid deformation with a 45 degree rotation and a translation vector of <0.5*width, 0.5*height>.

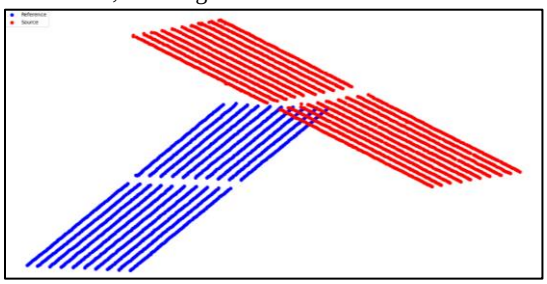

Figure 9: Orchard 2 with a combined rotational and translational deformation

4.2.2 Computation time

To evaluate the difference in computational complexity of each method, each of the accuracy tests in 4.2.1 will have the total computation time of the registration recorded in seconds.

4.2.3 Robustness

The robustness of each method to unexpected image distortions will be evaluated in the following ways:

- **Section removal**: Entire sections of the source point cloud will be removed before registering the clouds. This will model the cases where entire sections of trees were destroyed or omitted from capture by the drone during orthomosaic generation. This will be simulated through removing a quadrant of points in the cloud. Registration will then occur using the combined rotational and translation deformation parameters from 4.2.1 and the resulting accuracy will be recorded.

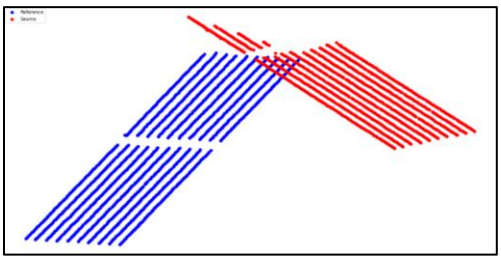

Figure 10: Orchard 2 with a rigid deformation and missing quadrant

- **Point removal by confidence level:** Each algorithm's robustness to missing points will be evaluated. Points will be removed based on their confidence levels. Low confidence detections are more likely to be false detections and outliers, and thus are unlikely to be present in subsequent surveys of that orchard. All accuracy tests until now have used all detections regardless of confidence level. This test will remove detections based on their confidence level, increasing the threshold in 10% intervals, with any points having a confidence level lower than the threshold being removed. Registration will occur (using the deformation parameters from 4.2.1) with the filtered input data, and the results will indicate each algorithm's robustness to missing low-confidence points. This test will also show whether the removal of low confidence points has any positive affect on registration accuracy.

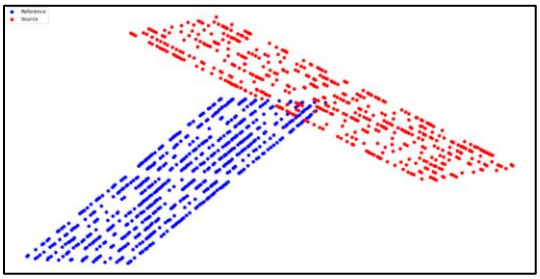

Figure 11: Orchard 2 with a 0.7 confidence level threshold and rigid deformation

- **Noise:** Each algorithm's robustness to additional outliers will be evaluated. This test will model the cases of false tree detections being produced by the feature extractor, which don't have a true counterpart in the previous survey's point cloud. Additional outliers will be added as

Guassian noise to the source dataset. For each point p in the source cloud of size n, an additional point taken randomly from a Guassian distribution of mean p and variance 10 will be added. This means each dataset will have a minimum of n outliers added to the source point cloud. Registration will then occur using the combined rotational and translation deformation parameters from 4.2.1 and the resulting accuracy will be recorded.

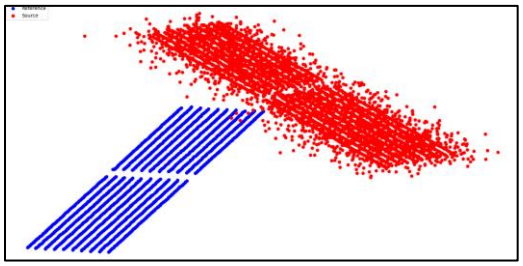

Figure 12: Orchard 2 with a rigid deformation and noise

4.3 Registration Error Metrics

The registration accuracy of each algorithm will be evaluated using 2 error metrics: Registration MSE and Human MSE.

4.3.1 Registration MSE

The error metric being used to determine the accuracy of a registration will be the Mean Squared Error (MSE) of the distance between closest point pairs in the transformed source and reference point cloud. Where the transformed source cloud is the result of the input source point cloud being transformed to align with the reference by the registration method. Calculated as follows:

$$\text{MSE} = \frac{1}{n}\sum_{k=1}^{n}(x_k - y_k)^2$$

Where:

- n is the number of points in the transformed source point cloud
- x_k is a point in transformed source point cloud
- y_k is the closest point to x_k in the reference point cloud (nearest neighbour)

Note that it's impossible for the MSE to be zero due to the presence of outliers in the dataset i.e., there are some points in the source cloud with no true counterpart in the reference and vice versa. Therefore, in order to contextualize these MSE results, a Base MSE will be calculated and compared to the registration MSE. This Base MSE is the preexisting error between the pre-registered source and reference point clouds, exactly as provided by Aerobotics. This relates to the registration error of their current human-based approach. Assuming the provided human registration result was the optimal solution, this base MSE would indicate the inherent error in the dataset due to outliers. And any differences in the Registration MSE and Base MSE error would indicate the relative registration accuracy levels of the human and PCR methods. This metric will be referred to as ICP/CPD MSE in the tabulated results.

4.3.2 Human MSE

The accuracy of each registration method will also be compared to the current human-based method at Aerobotics. This will be done through calculating the MSE between the transformed source cloud and the original, pre-registered source cloud provided by Aerobotics. If these PCR results match the human registration method exactly, a MSE value of 0 is possible as there are no outliers between the same point set. This will give an indication of how closely these methods compare to human-based approaches. This metric will be referred to as ICP/CPD Human MSE in the tabulated results.

4.3.1 Qualitative Evaluation

Note that the case when the calculated Registration MSE is lower than the Base MSE does not necessarily mean that the PCR method is more accurate than the human method. This reduction in MSE could be due to the PCR method aligning more outliers together than true detections. Therefore, an additional evaluation method must be used to ensure the PCR method is registering the data as expected. Additional qualitative evaluation will occur through visual analysis of registration accuracy. The results of registration will be plotted for examination and judgement. This will provide visual confirmation that the method is registering the data accurately. This is the same type of evaluation done by Aerobotics' existing human registration method.

4.4 Software Implementation

The algorithms and their evaluation components were implemented in Python 3 [13]. In order to ensure the integrity of their implementation and avoid the possibility of unseen implementation errors skewing any results, established libraries were used to perform the registration. These libraries were all open source and under the MIT license. The "PyCPD" [14] library was used for the CPD implementation and the "ICP" [15] library was used for the ICP implementation. The open repositories were cloned, and the code was edited to meet the requirements for the evaluation of this project.

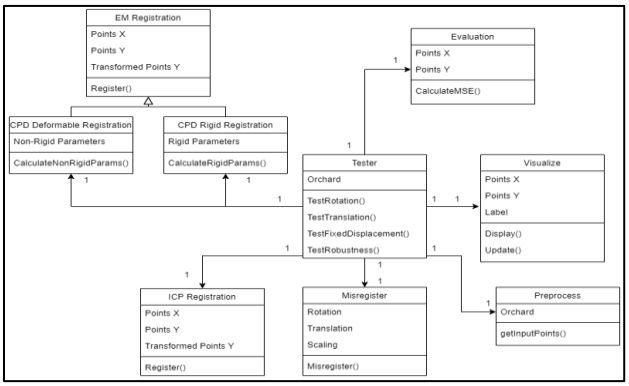

Figure 13: Class Diagram of Software System

Figure 13 shows the class diagram of the implemented software system. The main class is the Tester, which uses the methods in the other classes to evaluate the accuracy and robustness of each algorithm. The Misregister class contains functionality to misregister a given input cloud with given affine deformation parameters. This is necessary as the input data is pre-registered and thus has to be purposefully misregistered in order to evaluate the accuracy of each algorithm. The Preprocess class provides the functionality outlined in 3.1 and returns the projected centroids of the tree detections for both surveys of an orchard. The Visualize class is responsible for the visualization of the data and registration through point plots. The Evaluation class contains methods to determine the accuracy of a performed registration. The ICP registration class contains the methods relating to the ICP algorithm. The CPD Deformable and Rigid registration classes inherit from the EM registration class and contain methods relating to the CPD algorithm. These classes were taken and modified from the PyCPD library [14].

4.5 Hardware Specifications

All tests are run on a machine with the following specifications:

- Processor: Intel(R) Core (TM) i7-10700 CPU @ 2.90GHz
- RAM: 16GB DDR4 3200MHz

5 Results and Discussion

The results of the evaluation methods outlined in section 4, concerning both accuracy and robustness, are presented below for analysis and discussion.

5.1 Rigid Registration Accuracy Results

The results for the registration accuracy evaluations concerning rigid deformations are as follows:

5.1.1 Rotational Deformations

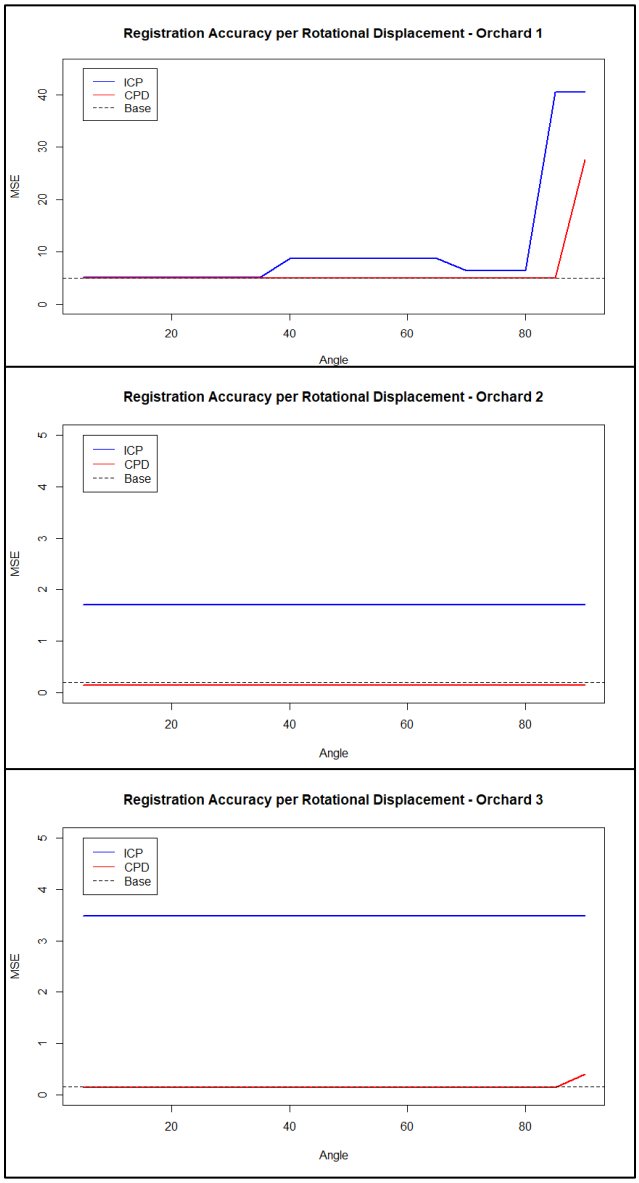

Figure 14: Registration Accuracy Results for Orchards 1-3 with varying degrees of Rotational Deformations

In all 3 datasets, the error of the CPD algorithm is well below the ICP error and as such is generally more accurate in cases of individual rotational deformations, as seen in Figure 14. Studying the graphs, it's clear that the result of the CPD algorithm is relatively more invariant to the degree of rotational deformation applied to the input source cloud. The only time the CPD error increases is in orchards 1 and 3 around a 90 degree rotation. This is likely due to the algorithm flipping the source cloud the opposite direction during the registration. This will likely not be an issue as in general the rotational differences between surveys will not exceed 30 degrees. For these reasons, it's clear that the CPD

algorithm is the better choice for datasets with rotational deformations under 90 degrees.

5.1.2 Translational Deformations

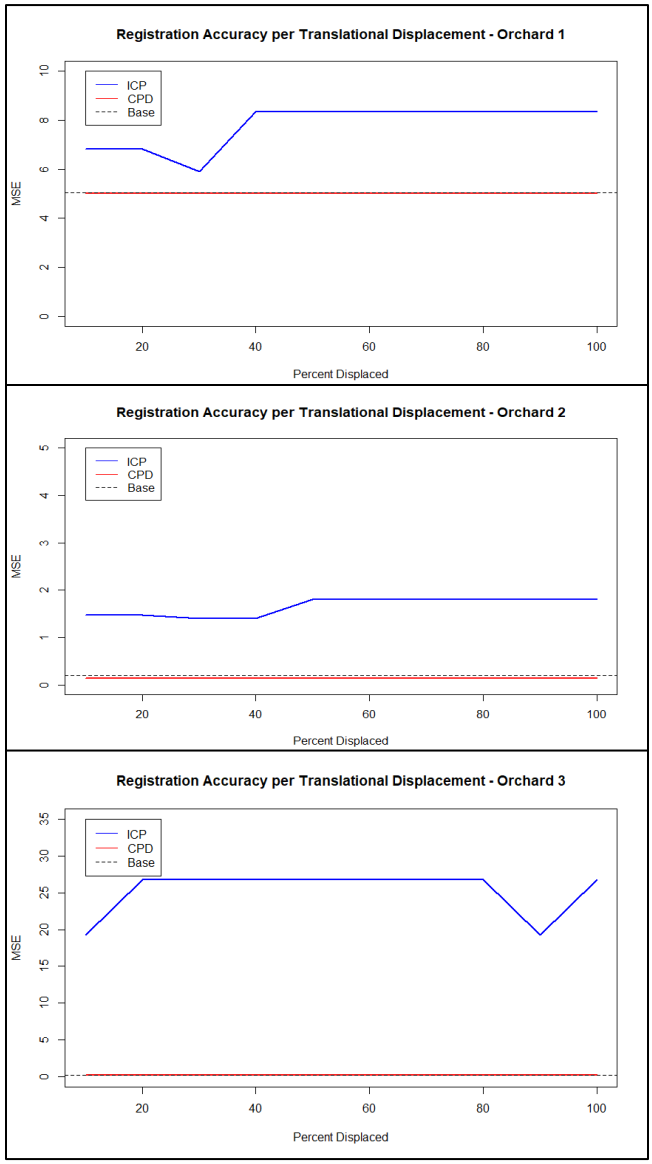

Figure 15: Registration Accuracy Results for Orchards 1-3 with varying levels of Translational Deformations

Studying the graphs from Figure 15, one can conclude that the result of a CPD registration is invariant to the degree of translational deformation applied to the input source cloud. This is to be expected as the first step of the rigid registration process for CPD is to center the source cloud on the reference [10], rendering itself robust to a large range of translational deformations. ICP, on the other hand, produces varying levels of registration accuracy for different levels of translational deformation. This is to be expected as different sets of closest point pair correspondences will be made based on the relative position of the 2 clouds. These different correspondences will result in varying transformation parameters being estimated, resulting in inconsistent levels of registration accuracy over different translation deformations. For this reason, and the fact that the CPD error results closely match the human method's Base MSE, it's clear that the CPD algorithm is better suited to datasets with translational deformations.

5.1.3 Full Rigid Deformations

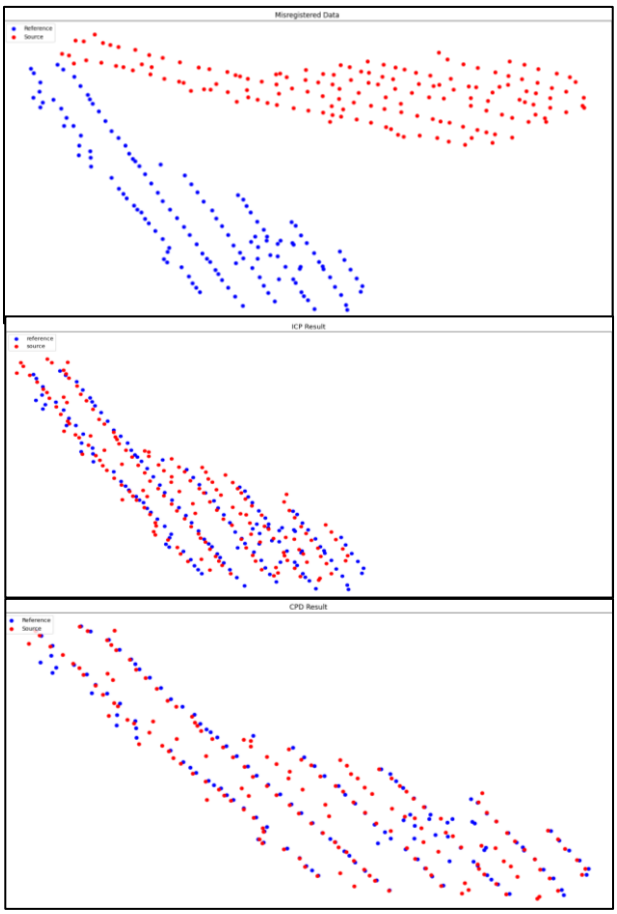

Figure 16: Visual registration results for CPD (bottom) and ICP (middle) on misregistered orchard 1 data (top)

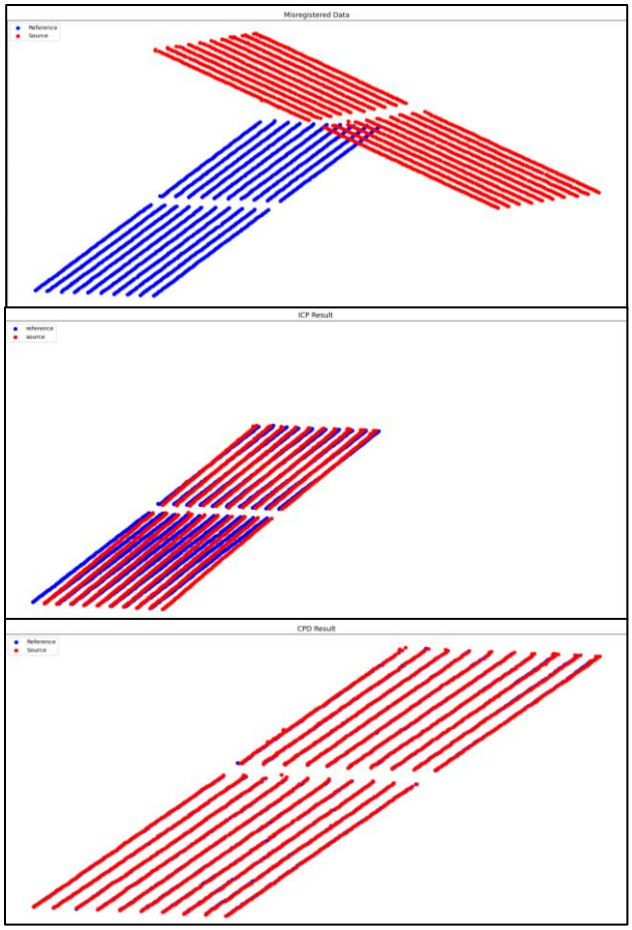

Figure 17: Visual Registration results for CPD (bottom) and ICP (middle) on misregistered orchard 2 data (top)

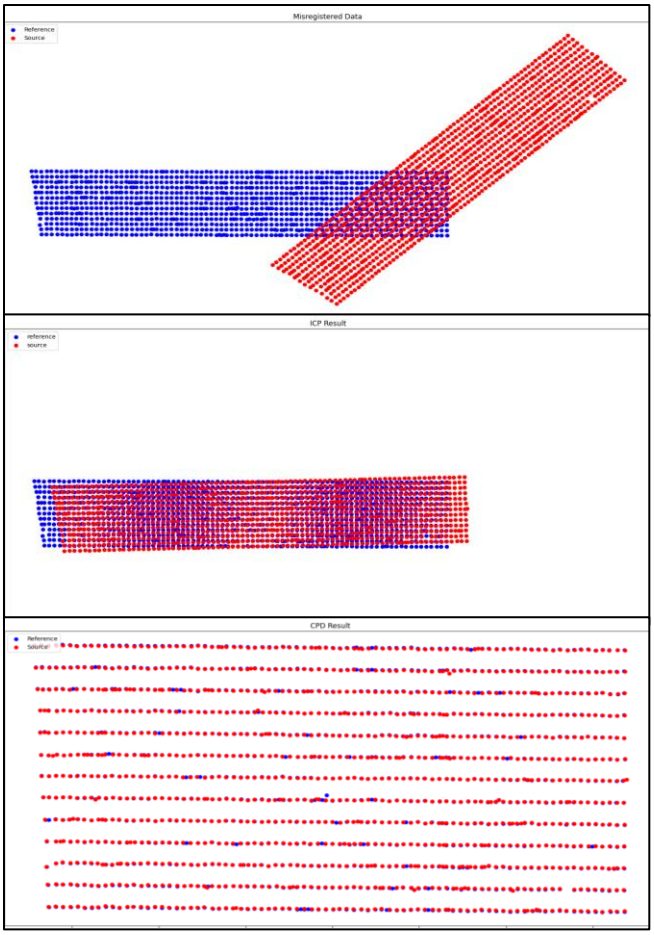

Figure 18: Visual registration results for CPD (bottom) and ICP (middle) on misregistered orchard 3 data (top)

Orchard	ICP MSE	CPD MSE	Base MSE	ICP Human MSE	CPD Human MSE
1	8.33	5.02	5.05	6.54	0.06
2	1.37	0.14	0.19	1.32	0.16
3	9.32	0.14	0.15	9.13	0.01

Table 2: Registration accuracy results of each algorithm with a rigid deformation

As seen in Figures 16-18, the CPD algorithm appears to be more accurate than ICP through a visual examination. The result for orchard 1 appears to be comparable, however this is likely due to the fact that it has the least number of data points and as such ICP is less likely to form incorrect point correspondences and therefore will result in less error. The ICP results for orchards 2 and 3 are visibly inaccurate. This is due to the well-known problem with ICP registration called *local minima convergence* [12]. ICP works by

estimating transformation parameters that minimize an error metric (average distance between closest point pairs). A round of ICP registration only guarantees a local minimum in registration error from that particular initialization point (initial position of the source cloud). This is illustrated in the tests on translation deformation in Figure 15, the resulting registration error from ICP differs based on the starting point of the source cloud. It's clear in Figures 17 and 18 that the ICP algorithm has fallen in to local minima traps and is thus unable to accurately register the point clouds (or find the global minima. CPD on the other hand appears to align the point clouds such that the transformed source perfectly overlays the reference in Figures 16-18. It seems to have found the global minimum in registration error, unlike ICP. The results in table 2 support this claim. The error results for CPD are far lower than that of ICP consistently. They are even lower than that of the Base MSE (which is the error of the human-based registration). This means either CPD provides a more accurate registration than the previous human based method, or it has aligned more outliers with points, resulting in a lower MSE overall. The Human MSE results indicate how close the algorithms came to the human registration. The CPD human MSE errors are consistently lower than the ICP, indicating it is closer to matching the human based approach. In the case of orchard 3, it appears that the CPD method almost exactly matches the human registration method with a Human MSE of 0.01. This is likely explained by the fact that Orchard 3 has the highest overall average detection confidence levels (Table 1), meaning there are less outliers in the dataset that harm registration accuracy due to false point correspondences.

For these reasons, it's clear that the CPD algorithm outperforms ICP in terms of registration accuracy for data with rigid deformations. Note that the simulated deformations used for testing here are far more severe than the real deformations encountered by Aerobotics with the raw, unregistered input data. The real deformations are caused by errors in the GPS tagging of the drone capturing the images during orthomosaic generation and are thus minor. These more significant deformations were used to test the limits of each algorithm.

5.2 Computation Time Results

Orchard	ICP Time	CPD Time	ICP Iterations	CPD Iterations
1	0.02	0.04	18	45
2	2.5	55.55	210	186
3	0.26	6.49	43	88

Table 3: Computation time of each algorithm in seconds for the registration of the full rigid deformation in 5.1.3

As seen in Table 3, the increased registration accuracy provided by the CPD comes at an equally significant increase in computation time, relative to ICP. In most cases, the CPD algorithm has more iterations than ICP, except for orchard 2. However, orchard 2 also has the most points of the three orchards. Therefore, it requires a lot more computation for the CPD algorithm, explaining the higher computation time. If orchard 2 represents an upper bound on the size of the datasets processed by Aerobotics, then both algorithms register the datasets faster than the current human method. With CPD providing a sufficiently accurate registration, compared to the human method, in a fraction of the time (the human method is estimated to take an average of 5 minutes based on the training videos provided by Aerobotics).

5.3 Robustness Results

The results of the methods evaluating the robustness of each algorithm to unforeseen deformations, as outlined in section 4.2.3. These were tested using the same rigid deformations used in section 5.1.3.

5.3.1 Section Removal:

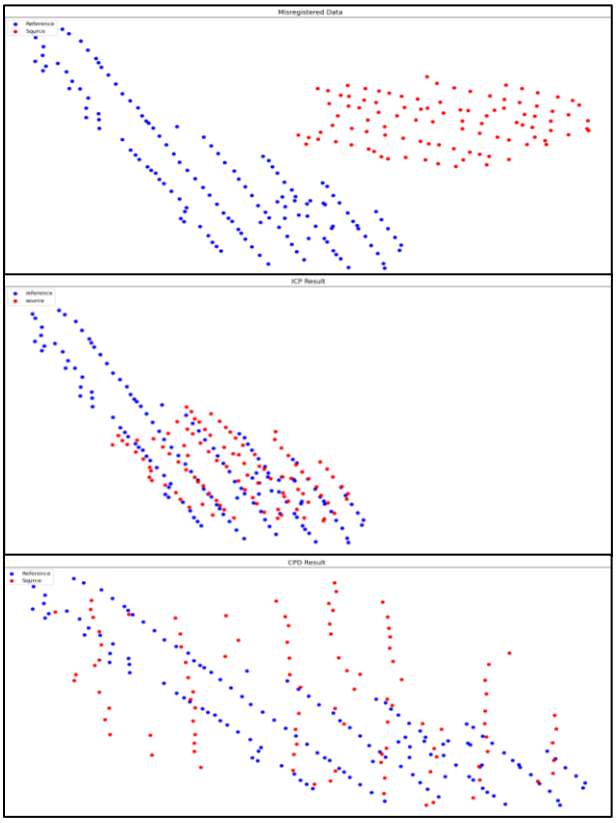

Figure 19: Visual registration results for CPD (bottom) and ICP (middle) on misregistered orchard 1 data with missing section (top)

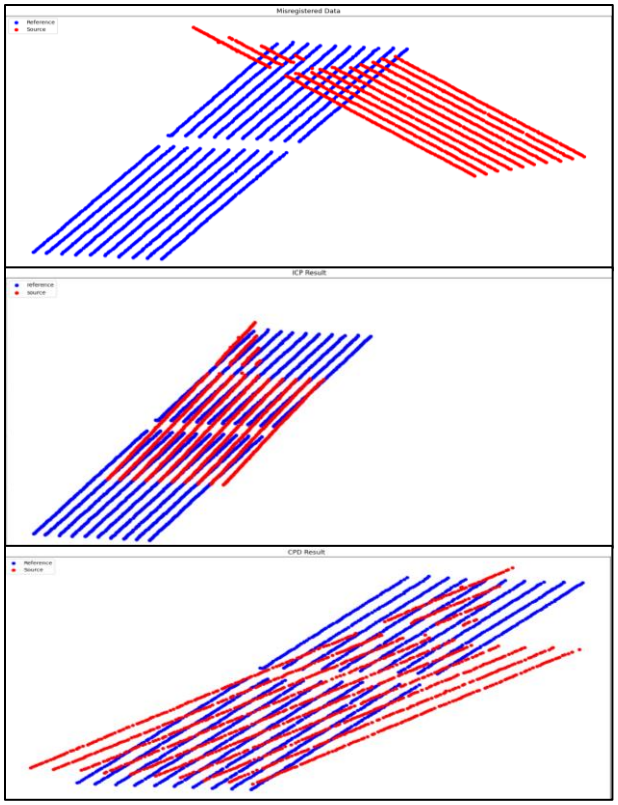

Figure 20: Visual registration results for CPD (bottom) and ICP (middle) on misregistered orchard 2 data with missing section (top)

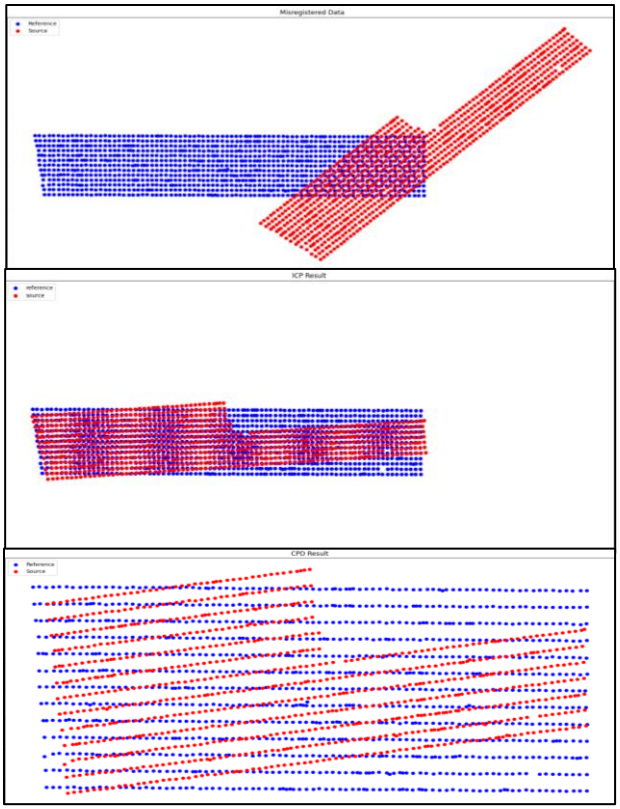

Figure 21: Visual registration results for CPD (bottom) and ICP (middle) on misregistered orchard 3 data with missing section (top)

Orchard	ICP MSE	CPD MSE	Base MSE	ICP Human MSE	CPD Human MSE
1	6.28	196.55	5.11	4.34	659.64
2	2.67	21.31	0.20	214.29	222.49
3	4.26	4.17	0.15	23.35	19.96

Table 4: Registration accuracy results of each algorithm on missing section data

Analyzing Figures 19-21, it's clear that no method provides an accurate registration on data with a missing section in the source cloud. ICP comes close in orchard 1 but is clearly misaligned in orchards 2 and 3. CPD displays even worse performance than ICP in most cases, especially in orchard 1. This is because the CPD algorithm estimates scaling transformation parameters as part of it's rigid registration parameters (see section 3.2). The relatively large number of lost points in orchard 1 prompt the algorithm to estimate significant scaling transformations during the registration process, when no scaling deformations were performed on the input data. This results in an extremely inaccurate registration, as evident by the high error rates for CPD in Table 4. Both algorithms fail to

accurately register the data. This is evident by the fact that their registration error levels are consistently above the Base MSE and shown by their high Human MSE levels. Therefore, both ICP and CPD are not robust to datasets with missing sections in one of the input clouds.

5.3.2 Point removal by confidence level

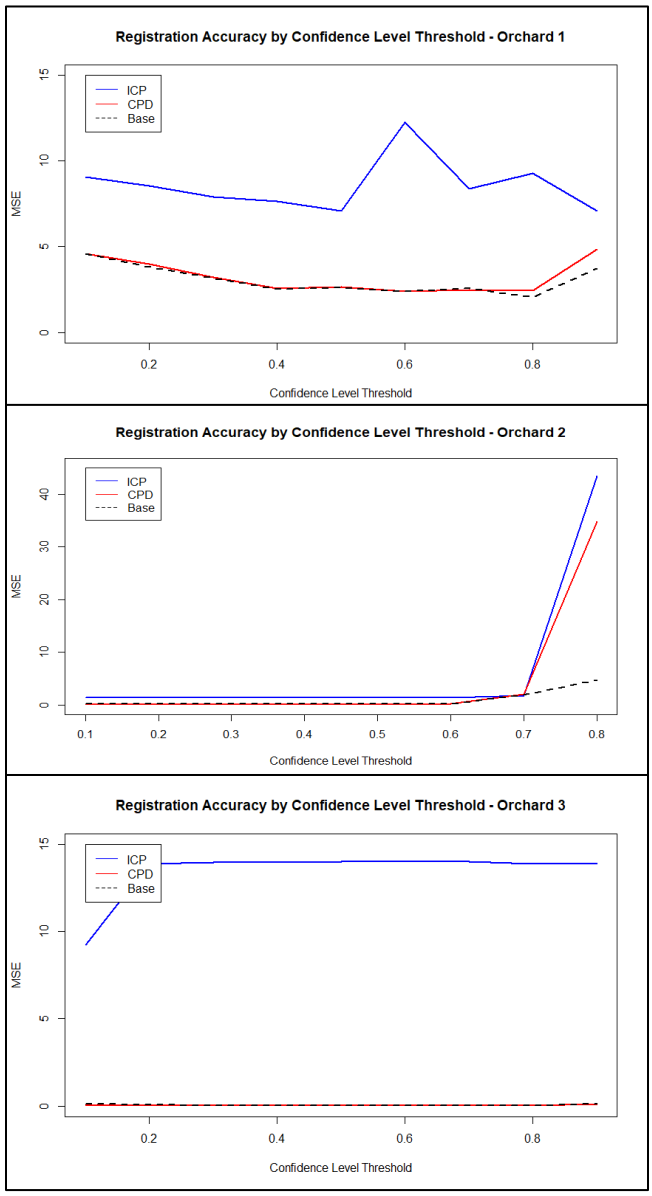

Figure 22: Registration accuracy results for orchards 1-3

Figure 22 shows the registration accuracy results of each algorithm with input points above a certain confidence level threshold. This tests the hypothetical case where these low confidence detections were not picked up by the feature detector at all. This test aims to evaluate each algorithms robustness to datasets with missing low confidence detections. This was tested to the extreme, using a threshold up to 90 percent (except in the case of orchard 2 as a 90 percent threshold didn't provide enough input points to run any algorithm). Analyzing the graphs, it's clear that increasing the confidence threshold has a minimal effect on registration accuracy. ICP accuracy remains relatively constant over the confidence thresholds, curiously becoming less accurate over the 20 percent threshold in orchard 3. CPD remains relatively constant over increasing thresholds too, with minimal perceived change in orchards 2 and 3 and a reduction in error in orchard. It appears that the removal of low confidence detections improves CPD registration accuracy in orchard 1 (before the 0.8 threshold). Both ICP and CPD have a sharp increase in orchard 2 around the 80 percent threshold. This is likely due to the removal of a large number of true detections, resulting in too few points as input to produce an accurate registration. Orchard 3 has the highest overall mean confidence level of detections (over 90 percent), therefore the increase in confidence level threshold would not remove nearly as many points as in the other 2 orchards. This explains it's relatively constant level of registration accuracy. It's unlikely that any detections under a 50 percent confidence level would ever fail to be detected by the feature extractor. Each method shows constant level or even a reduction in registration error from thresholds 0.1 to 0.5 (except ICP in orchard 3 which has a slight increase). Therefore, both algorithms are robust to cases where low confidence points are missing from the input data, and in fact may show an increase in registration accuracy in these cases.

5.3.3 Noise

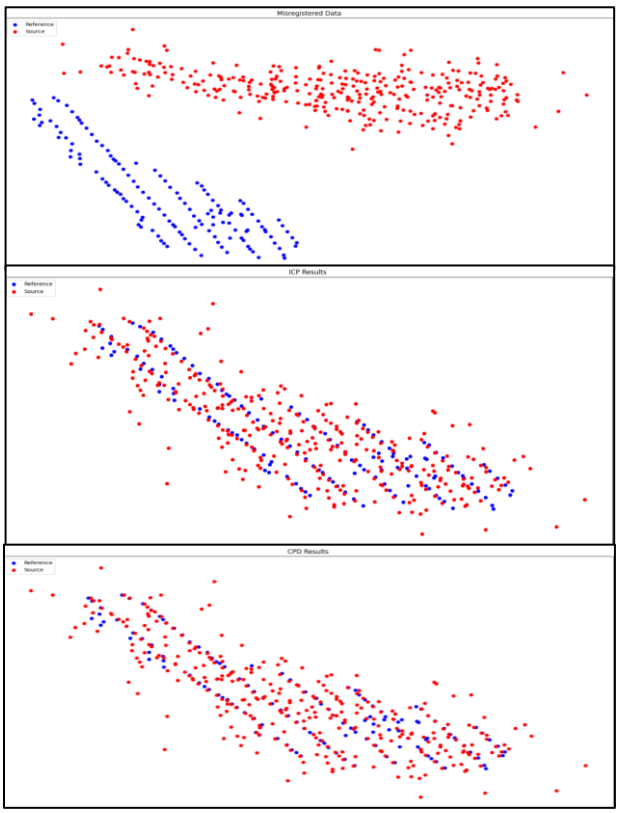

Figure 23: Visual registration results for CPD (bottom) and ICP (middle) on misregistered orchard 1 data with noise (top)

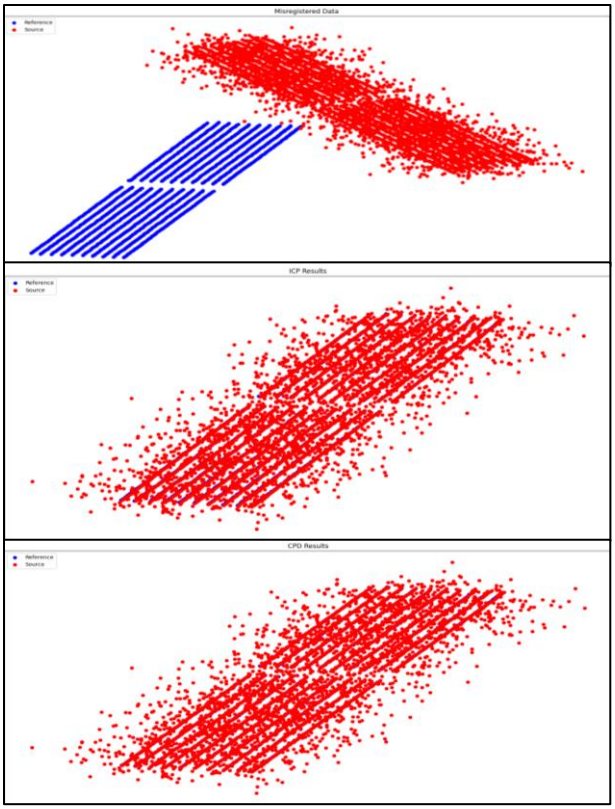

Figure 24: Visual registration results for CPD (bottom) and ICP (middle) on misregistered orchard 2 data with noise (top)

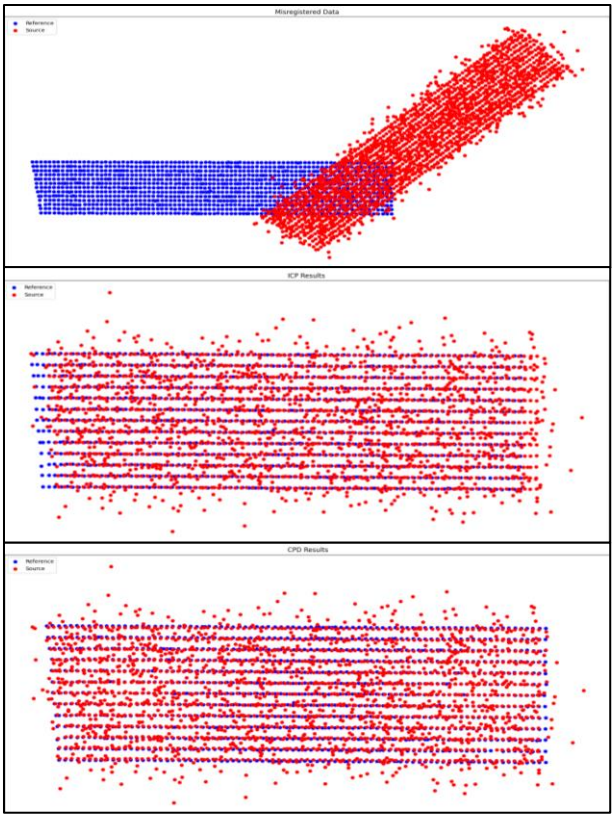

Figure 25: Visual registration results for CPD (bottom) and ICP (middle) on misregistered orchard 3 data with noise (top)

Orchard	ICP MSE	CPD MSE	Base MSE	ICP Human MSE	CPD Human MSE
1	27.39	27.66	27.27	4.45	0.17
2	12.65	12.70	12.74	0.27	0.20
3	12.66	7.13	7.88	4.29	1.72

Table 5: Registration accuracy results of each algorithm on data with noise

Figures 23-25 show the visual results of each registration method being applied to datasets in which additional outliers have been added using Guassian noise. This models the case where the feature extractor makes erroneous false detections in later orchard surveys or detects trees that are new or were missed in the previous survey. The extreme case presented here, where half the points in the source cloud are simulated outliers, is used to test the limits of each algorithms' robustness to noise/outliers.

Visually, analyzing Figures 23-25, it appears each method has accurately registered the dataset. This is especially surprising in the case of ICP, a method well known to be sensitive to noise and outliers [8, 9, 12]. ICP appears to provide an accurate registration in all cases, except orchard 3 where the source cloud appears to be

slightly misaligned to the right of the reference cloud. Analyzing the results in Table 5, it's clear that the ICP registration errors are much closer the Base MSE levels than in Table 2, where the same test occurred on data without the additional noise. ICP also presents lower Human MSE levels with this dataset than the dataset without the noise (in which the same rigid deformation was applied). This shows that the ICP method performed better on the noisy dataset than the original. Visually, the CPD results seem similar in accuracy to ICP (except in the case of orchard 3 where CPD displays a slight improvement in accuracy). Analyzing the error results in Table 5, it's shown that ICP provides a lower registration error than CPD in orchards 1 and 2. At first glance this would suggest ICP outperforming CPD on those datasets, however CPD shows significantly lower Human MSE's than ICP in those cases. Meaning it's closer to the human registration method and thus likely more accurate as the human method has been visually evaluated to be correct by a professional. CPD is shown to provide a more accurate registration in the case of orchard 3, with a lower registration and Human MSE, as expected after the visual comparison. However, there is still a slight misalignment after the CPD registration for orchard 3, as indicated by the relatively high Human MSE metric (despite the Registration MSE being lower than the Base MSE). This slight misalignment is visible in Figure 25. Unlike ICP, the registration accuracy of CPD results decreased with this dataset. This can be seen in the increase of CPD error results from Table 2 to Table 5, especially in the case of orchard 3 where the Human MSE jumped from 0.01 to 1.72. However, the accuracy of CPD is still superior to ICP on this data, determined by a visual comparison of their results and error metrics, and thus CPD is shown to be more robust to additional noise/outliers than ICP.

6. Extensions to the Algorithms

Proposed extensions to each algorithm, to improve registration accuracy and their suitability to the problem, are presented below.

ICP Improvement:

The registration accuracy and robustness results of the ICP algorithm make it clear that the basic ICP method used is not suitable to register the orchard input data accurately. This is due to the ICP algorithm only guaranteeing a result with a local minima of registration error, based on the starting position or initialization of the input point clouds during registration. Alterations and improvements need to be made to the algorithm for it to be a viable solution to this orchard registration problem. One such improvement is a method called Global Iterative Closest Point using Nested Annealing [12]. This method aims to avoid the local minima traps inherent to the ICP method through a doing a search for the global minima, or true registration. This is completed through multiple rounds of ICP registration being done on the data, with different initializations. The local error is recorded for each round, and the algorithm terminates when the global minimum is found, resulting in an accurate registration. The implementation and evaluation of this proposed extension to the ICP algorithm is left as future work.

CPD Improvement:

The CPD algorithm has a few inherent shortcomings: It's not guaranteed to converge on every run, the registration transformation parameters for motion coherence produced by the algorithm are unclear and unintuitive, the algorithm is sensitive to shape rotation (as seen in section 5.1.1) and the acceleration of the algorithm is restricted to the use of the Guassian kernel [16]. A new variation of the CPD algorithm, called Bayesian Coherent Point Drift (BCPD), was introduced in 2020 [16]. This algorithm overcomes these issues with traditional CPD on the basis of variational Bayesian inference. This new algorithm has the following characteristics:

- It guarantees convergence.
- It provides intuitive registration parameters regarding motion coherence.
- It combines rigid and non-rigid registration in to a single algorithm thereby enhancing robustness to target rotation.
- The algorithm can be accelerated with non-Guassian kernel functions, greatly improving CPD computational efficiency.

These improvements offered by this newer CPD algorithm are likely to result in greater registration accuracy levels than the base method implemented in this project. The implementation and evaluation of this proposed extension to the CPD algorithm is left as future work.

Non-rigid registration:

The results presented and discussed in this paper all regard two rigid registration algorithms. As such, they were only tested on data with rigid deformations (translation and rotation) applied to them. Rigid registration algorithms estimate transformation parameters that preserve the distance between every pair of points, whereas non-rigid registration algorithms have no restriction. The case/s where the input data displays any non-rigid deformations were not tested in this project. This was due to the input data for this project being pre-registered, as such any of the real-world deformations that occur between the orchard detections of successive surveys were removed. Information about the typical instances of deformation present in the raw input data was lacking, therefore only rigid deformations were simulated. The base ICP method implemented in this project is only capable of rigid registration, however the CPD method contains an algorithm for the non-rigid registration too. In fact, CPD is considered a state-of-the-art non-rigid registration algorithm due to it's registration performance and scalability to large points sets [17]. As such, it is expected to be able to accurately register the provided input data with non-rigid deformations. The BCPD algorithm described above is also capable of non-rigid deformations and, unlike base CPD, has combined the non-rigid and rigid registration algorithms into a single method capable of handling both types of registration. Extensions to the ICP method, capable of non-rigid point set registration, have been developed [18]. These methods have been shown to find the

optimal non-rigid deformation on the input data iteratively, while maintaining the convergence properties of the original ICP algorithm. The implementation and evaluation of these methods is left as future work.

Conclusions

The aim of this paper was to determine whether the problem of orchard orthomosaic registration at Aerobotics, using tree detection features, could be modelled as a PCR problem. And whether two types of rigid PCR methods (ICP and CPD) were capable of accurately registering the input tree detections, modelled as point clouds. The methods must produce results comparable to the accuracy of the existing human-based registration method to be deemed acceptable. The two methods chosen take significantly different approaches to PCR, with ICP making use of closest-point correspondences to estimate registration parameters and CPD maximizing the probability of overall cloud alignment to register the input point clouds. This difference in registration methodology between the methods is accompanied by an equally significant difference in the results produced by each method. ICP is shown to produce relatively inaccurate registration results, not coming close enough to the accuracy of the existing human method to be considered a viable approach to orchard orthomosaic registration at Aerobotics. CPD is shown to outperform ICP in all the tests over rigid registration accuracy, albeit at a higher computational cost. Additionally, CPD is shown to be more robust to various unforeseen deformations of the input data, including the addition of noise to the dataset and the removal of low confidence detections. It's shown that CPD produces qualitative (visual registration results) and quantitative (human and registration error metrics) results comparable to the existing human-based registration methods at Aerobotics. Taking the human registration results as the standard of an acceptable registration, we can therefore conclude that CPD is capable of an accurate rigid registration of the input orchard data and thus meets the aim of the project in the rigid registration case. However, we can't conclude that CPD is fully capable of accurately registering all input orchard data, as the cases where the input data contains non-rigid deformations were not tested. We were unable to obtain raw input data with these deformations present in the timeframe of this project and thus leave it to future work. Although, based on published experiments of CPD results on other datasets containing non-rigid deformations, it's expected that the CPD method will be capable of providing an accurate level of registration accuracy on cases of non-rigid deformations in the orchard data [10, 16, 17].

REFERENCES

[1] Dalila Cervantes-Godoy and Joe Dewbre. 2010. Economic importance of agriculture for poverty reduction. OECD Food, Agriculture and Fisheries Working Papers, No. 23, OECD Publishing. DOI: https://doi.org/10.1787/5kmmv9s20944-en

[2] Mitchell C. Hunter, Richard G. Smith, Meagan E. Schipanski, Lesley W. Atwood, David A. Mortensen. 2017. Agriculture in 2050: Recalibrating Targets for Sustainable Intensification, *BioScience*, Volume 67, Issue 4, April 2017, Pages 386–391, DOI:https://doi.org/10.1093/biosci/bix010

[3] Tsouros DC, Bibi S, Sarigiannidis PG. A Review on UAV-Based Applications for Precision Agriculture. Information. 2019; 10(11):349. DOI: https://doi.org/10.3390/info10110349

[4] Sean Nordstrum. 2020. What is an Orthomosaic Map and How Does Mapping Benefit My Property?. Retrieved from: https://blog.dronebase.com/what-is-an-orthomosaic-map-and-how-does-mapping-benefit-my-property

[5] Barbara Zitová and Jan Flusser. 2003. Image registration methods: a survey. Image and Vision Computing, (Oct. 2003), Volume 21, Issue 11, 977-1000. DOI: https://doi.org/10.1016/S0262-8856(03)00137-9

[6] Aerobotics. 2021. Aerobotics: Intelligent tools for Agriculture. Retrieved from: https://www.aerobotics.com/.

[7] Huang, X., Mei, G., Zhang, J., & Abbas, R. (2021). A comprehensive survey on point cloud registration. arXiv preprint arXiv:2103.02690.

[8] Procházková, Jana & Martišek, Dalibor. (2018). Notes on Iterative Closest Point Algorithm.

[9] Shaoyi Du, Juan Liu, Chunjia Zhang, Jihua Zhu, Ke Li, Probability iterative closest point algorithm for m-D point set registration with noise, Neurocomputing, Volume 157, 2015, Pages 187-198, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2015.01.019/ .

[10] A. Myronenko and X. Song, "Point Set Registration: Coherent Point Drift," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 12, pp. 2262-2275, Dec. 2010, doi: 10.1109/TPAMI.2010.46.

[11] Reynolds, D. A. (2009). Gaussian mixture models. Encyclopedia of biometrics, 741, 659-663.

[12] Tao Ngoc Linh, Hasegawa Hiroshi, Global Iterative Closet Point Using Nested Annealing for Initialization, Procedia Computer Science, Volume 60, 2015, Pages 381-390, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2015.08.147/.

[13] Python. 2001. Retrieved from https://www.python.org/

[14] Python-CPD. 2018. Retrieved from https://github.com/siavashk/pycpd/

[15] Iterative Closest Point. 2019. Retrieved from https://github.com/richardos/icp/

[16] O. Hirose, "A Bayesian Formulation of Coherent Point Drift," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 7, pp. 2269-2286, 1 July 2021, doi: 10.1109/TPAMI.2020.2971687.

[17] B. Maiseli, Y. Gu, and H. Gao, "Recent developments and trends in point set registration methods," Journal of Visual Communication and Image Representation, vol. 46, pp. 95–106, 2017

[18] B. Amberg, S. Romdhani and T. Vetter, "Optimal Step Nonrigid ICP Algorithms for Surface Registration," 2007 IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1-8, doi: 10.1109/CVPR.2007.383165.

[19] K. Sharma and A. Goyal, "Classification based survey of image registration methods," 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2013, pp. 1-7, doi: 10.1109/ICCCNT.2013.6726741.

[20] Saxena, S., & Singh, R. K. (2014). A survey of recent and classical image registration methods. International journal of signal processing, image processing and pattern recognition, 7(4), 167-176.

[21] Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60, 91–110 (2004). DOI: https://doi.org/10.1023/B:VISI.0000029664.99615.94/.

[22] Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). Computer vision and image understanding, 110(3), 346-359. DOI: https://doi.org/10.1016/j.cviu.2007.09.014/.

[23] A. Rangarajan, H. Chui, E. Mjolsness, L. Davachi, P. S. GoldmanRakic, and J. S. Duncan, "A robust point matching algorithm for autoradiograph alignment," MIA, vol. 1, no. 4, pp. 379–398, 1997.

[24] B. Luo and E. R. Hancock, "Structural graph matching using the em algorithm and singular value decomposition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 10, pp. 1120–1136, 2001

[25] S. Gold, C. P. Lu, A. Rangarajan, S. Pappu, and E. Mjolsness, "New algorithms for 2D and 3D point matching: Pose estimation and corresponding points." in NIPS, vol. 7. The MIT Press, 1995, pp. 957–964

[26] H. Chui and A. Rangarajan, "A new algorithm for non-rigid point matching," in CVPR, vol. 2. IEEE Press, Jun. 2000, pp. 44–51.

[27] B. Luo and E. R. Hancock, "A unified framework for alignment and correspondence," CVIU, vol. 92, no. 1, pp. 26–55, 2003.

[28] G. McNeill and S. Vijayakumar, "A probabilistic approach to robust shape matching," in IEEE ICIP, 2006, pp. 937–940.

[29] M. Sofka, G. Yang, and C. V. Stewart, "Simultaneous covariance driven correspondence (CDC) and transformation estimation in the expectation maximization," in CVPR, Jun. 2007

[30] Li, S., Wang, J., Liang, Z., Su, L.: Tree point clouds registration using an improved ICP algorithm based on kd-tree. In Proceedings of 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). 4545-4548.