UNIVERSITÄT FÜR BODENKULTUR WIEN
University of Natural Resources
and Life Sciences, Vienna

Master Thesis

Development and evaluation of algorithms for the automatic marker-free registration of forest point clouds obtained from Personal Laser Scanning

submitted by

Sarah WITZMANN, BSc

in the framework of the Master programme

Forstwissenschaften

in partial fulfilment of the requirements for the academic degree

Diplom-Ingenieurin

Vienna, October 2022

Supervisor:

Univ.Prof. Dipl.-FW. Dr. Arne Nothdurft
Institute of Forest Growth
Department of Forest- and Soil Sciences

Co-supervisors:

Dr. Tim Ritter BSc. MSc.
Dipl.-Ing. Christoph Gollob

Dedicated to Ralf Kraßnitzer, our hardworking technician, who hopefully never again has to manually co-register laser scans.

Affidavit

I hereby declare that I have authored this master thesis independently, and that I have not used any assistance other than that which is permitted. The work contained herein is my own except where explicitly stated otherwise. All ideas taken in wording or in basic content from unpublished sources or from published literature are duly identified and cited, and the precise references included.

I further declare that this master thesis has not been submitted, in whole or in part, in the same or a similar form, to any other educational institution as part of the requirements for an academic degree.

I hereby confirm that I am familiar with the standards of Scientific Integrity and with the guidelines of Good Scientific Practice, and that this work fully complies with these standards and guidelines.

Vienna, 20th November 2022

Acknowledgements

I would like to thank everyone who supported me during the completion of this master thesis and who motivated me in the past years. Special thanks go to my friend and colleague Laura, who made university life a pleasure and never failed to warn or remember me of deadlines or times for applications.

Above all, I would particularly like to thank the whole team of the Institute of Forest Growth with my supervisors Arne Nothdurft, Tim Ritter and Christoph Gollob leading the way. They never omitted encouraging me during my work and always provided assistance when needed.

Thanks to my whole family for the support and love they gave me and for always providing a place of refuge when I needed a break from the city.

On a final note, I want to thank my partner Michael, always backing me up and finding the perfect R-function for every situation.

Table of content

Abstract

The usage of data obtained from Personal Laser Scanning (PLS) for forest inventory purposes has increasingly gained recognition in the past few years. The advantages and thus the reasons for the growing popularity of PLS in forestry are not far to seek: Laser scanning technology allows for an acquisition of highly precise individual tree and stand information in a relatively short amount of time.

Nevertheless, the high labor cost efficiency of Personal Laser Scanning has one limitation which is not to be underestimated when “wall-to-wall” data, in contrast to forest inventory data with realtively small sample plots, is desired: larger areas (>1-3 ha) cannot be scanned at one go, due to the scanning time limitation associated with software and data processing restrictions. Therefore, multiple scans must be conducted and co-registered to obtain one continuous point cloud. One way of doing this is to use easily recognizable artificial reference markers, like white spheres placed on tripods. These markers can afterwards be detected by semi-automatic software and used for the referencing of spatially adjacent scans. However, the transportation and deployment of such markers constitute great logistic and occupational efforts. Taking into account the additional time needed for the preprocessing and the matching of the point clouds, this semi-automatic registration can be considered as bottleneck for the otherwise high efficiency of PLS-based inventory and monitoring on larger areas.

The goal of this master thesis was to develop and evaluate an algorithm for the automatic and marker-free registration of point clouds to eliminate this bottleneck and to pave the way for a more practical and time-efficient usage of PLS on larger areas in the future. 29 scans obtained from a forested area of 35 ha serve as data basis for this work. The point clouds were obtained with a GeoSLAM ZEB Horizon (GeoSLAM Ltd., Nottingham, UK), featuring a high data acquisition rate (300.000 points per second) and scanning range (100 m). Possible approaches for the registration algorithm, which will be tested in the thesis, include feature-based methods, like a rough 3D transformation using the tree positions and diameters as matching features, as well as individual 3D-point-based methods, directly matching point clouds based on the LiDAR data itself. The latter might be implemented using the Iterative Closest Point (ICP) method, which could serve as fine tuning after the rough registration of the point clouds. In summary, the aim of this work is to develop an easy-to-use algorithm for the automated, marker-free registration of forest point clouds and to evaluate the resulting point clouds in terms of their accuracy.

1. Introduction

Forest inventory plays an important role in the sustainable management of forest ecosystems. Stand parameters, such as the growing stock timber and the increment, were usually derived from aggregates of single-tree parameters like DBH and height that were used as input for taper functions. These stand-level estimates are crucial for the planning of thinning or harvests [1]. Through the introduction of Terrestrial Laser Scanning (TLS) for forest inventory purposes, the methods and possibilities of data acquisition for forest inventory purposes have undergone significant changes in the past two decades. The usage of TLS for forest mensuration does not only improve accuracy and efficiency, but also allows for an easier determination of stand volume and biomass as well as the conduction of repeated measures across time [2]. A further enhancement was achieved through the introduction of Personal Laser Scanning (PLS), allowing for an improved mapping and a higher labour efficiency [3].

However, the efficiency of PLS in forestry also has limitations: Due to the restricted scanning time associated with software and data processing, point clouds of larger areas can't be obtained in one go. Multiple scans of smaller subareas must be conducted, resulting in separate point clouds which are all produced in their own local coordinate reference systems. These separate point clouds need to be co-registered to merge the separate datasets and obtain one continuous point cloud of the study area. Usually, easily recognizable targets like white spheres are used as reference objects for a manual registration of point clouds [4]. Although a high registration accuracy can be achieved via this approach, its high time consumption and intensity of labour strongly restrict the practicability of the described method [5]. Therefore, the aim of this master thesis is to find and evaluate an algorithm for the automated, marker-free registration of multiple PLS point clouds.

Several methods for point cloud registration have already been developed and described in literature [6–10]. Apart from the already mentioned target-based methods, which need exterior information like GNSS (Global Navigation Satellite System) data [11] or reference targets [12], feature- and point-based methods are commonly used approaches. Feature-based methods use features which can be identified within the point cloud itself, working without any additional input [13–15]. Similarly, point-based methods directly match point clouds based on the LiDAR points themselves and on the geometric information they provide [16,17]. Such information might for example be generated by computing "spin-images", as described by Johnson [10]. Since these geometric descriptors are computed via the relative positions of the points to each other, this method is invariant to a changing translation of the point clouds [10]. A similar approach was described by Yang et al. [6], who took the relative distances and positions between separate points as input features to decouple scale, rotation and translation for an easier computation of the transformation. Figure 1 was adapted from a summary on existing registration methods by Guan et al. [16] and lists the different groups of co-registration as well as exemplary applications of them which can be found in literature.

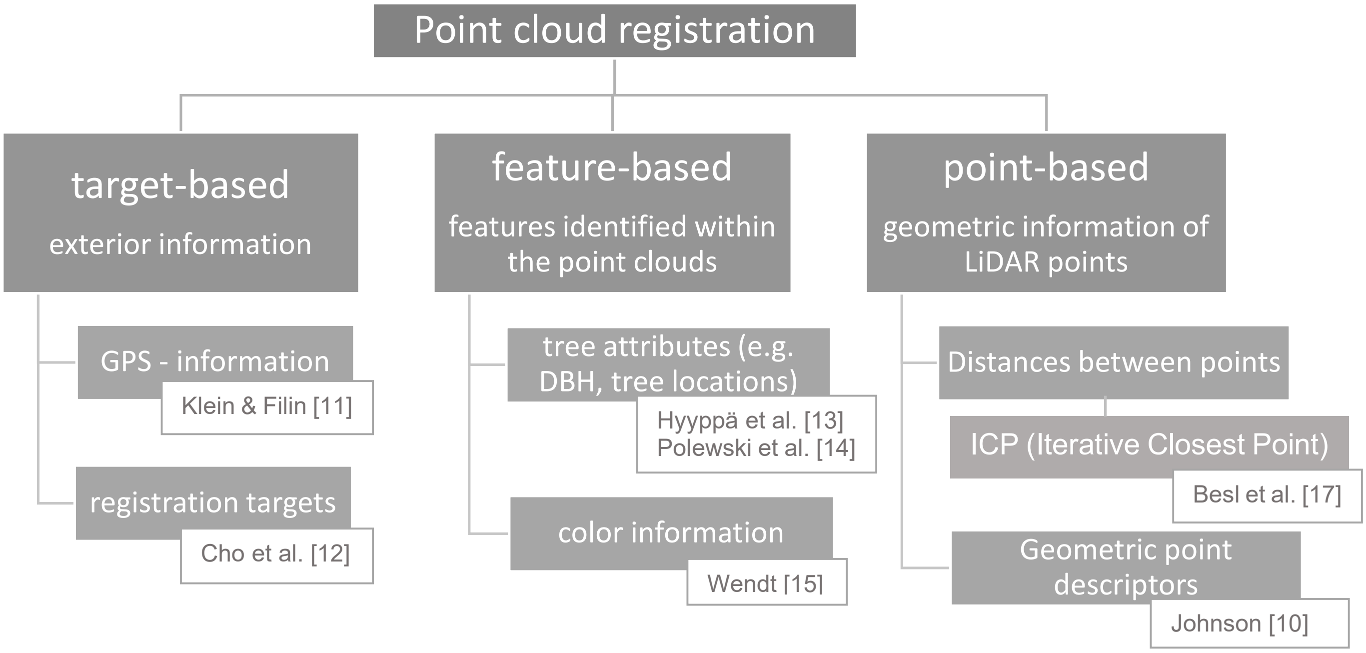


Figure 1: Illustration of different methods for point cloud registration. Adapted from Guan et al. [16].

The goal of this thesis was to adapt an algorithm described by Hyyppä et al. [13], which was initially developed to co-register the priorly extracted tree locations of terrestrial and airborne point clouds. In the course of this work, this algorithm was adapted in such a way that it allows for the coarse registration of the tree positions of multiple, partially overlapping terrestrial point clouds using R software (version 4.2.1).

2. Materials and Methods

Table 1 gives an overview of the materials and methods used in the course of this thesis, with detailed explanations following in the next sections.

Table 1: Workflow table

<table>
<tr><th>Step No.</th><th colspan="2">Step/ Sub-step</th><th>Hard-/ Software</th></tr>
<tr><td>1</td><td colspan="2">Scanning</td><td>GeoSLAM ZEB Horizon</td></tr>
<tr><td>2</td><td colspan="2">Preprocessing of the point clouds</td><td>Workstation / GeoSLAM Hub</td></tr>
<tr><td>3</td><td colspan="2">Tree detection and extraction of tree locations as input data</td><td rowspan="15">Workstation / R & R Studio</td></tr>
<tr><td>4a</td><td rowspan="3">Computation of feature descriptor vector for each tree</td><td>Spanning of four quadrants (one axis in direction of nearest neighboring tree, one axis perpendicular to the other) around each tree location</td></tr>
<tr><td>4b</td><td>Identification of the nearest neighboring tree in each of the four quadrants within a search radius of 10 m</td></tr>
<tr><td>4c</td><td>Computation of angles and distances to the nearest neighboring trees (8-digit feature descriptor vector)</td></tr>
<tr><td></td><td colspan="2">Computation of Euclidean distances between the feature descriptors of the two datasets</td></tr>
<tr><td></td><td colspan="2">Ranking matching pairs (Euclidean distance below threshold) based on second nearest neighbor distance ration (NNDR)</td></tr>
<tr><td>5c</td><td rowspan="3">Identification of optimal parameter k</td><td>Selection of k = 60 best matching pairs and calculation of the corresponding transformation parameters</td></tr>
<tr><td>5d</td><td>Computation of variance between the calculated y-translations in steps of 3</td></tr>
<tr><td>5e</td><td>Selection of optimal k (median of the triplet with smallest variance)</td></tr>
<tr><td>6a</td><td rowspan="4">Computation of the transformation parameters</td><td>Application of the k transformations to all points of the point pattern which must be transformed</td></tr>
<tr><td>6b</td><td>Calculation of the number of matching pairs (Euclidean distance below threshold) for each of the k transformations</td></tr>
<tr><td>6c</td><td>Selection of the transformation which results in the highest number of matching pairs</td></tr>
<tr><td>6d</td><td>Iterative optimization algorithm for final adjustment of selected transformation parameters</td></tr>
</table>

2.1. Data acquisition

The study site was located nearby Jaidhof (Lower Austria) and covered an area of approximately 35 ha. To gain area-covering point clouds, 31 scans had to be conducted (Fig. 2). Since the Scans 19 and 21 did not contain enough trees for the registration algorithm to work, they could not be co-registered, involving the exclusion of Scans 22, 23 and 24. Thus, 26 scans spanning an area of approximately 30 ha were included for the development and testing of the co-registration algorithm. The device used for the scanning process was the Personal Laser Scanner (PLS) GeoSLAM ZEB Horizon (GeoSLAM Ltd., Nottingham, UK), featuring a scanning speed of up to 300.000 points per second. The portability and relatively small weight of the scanner facilitate the fast and flexible acquisition of large amounts of data. A minimum of 3 white spheres was deployed in the overlap areas of adjacent scans as reference points, allowing for an opportunity to compare the output of the registration algorithm with the results of a target-based, manual co-registration.

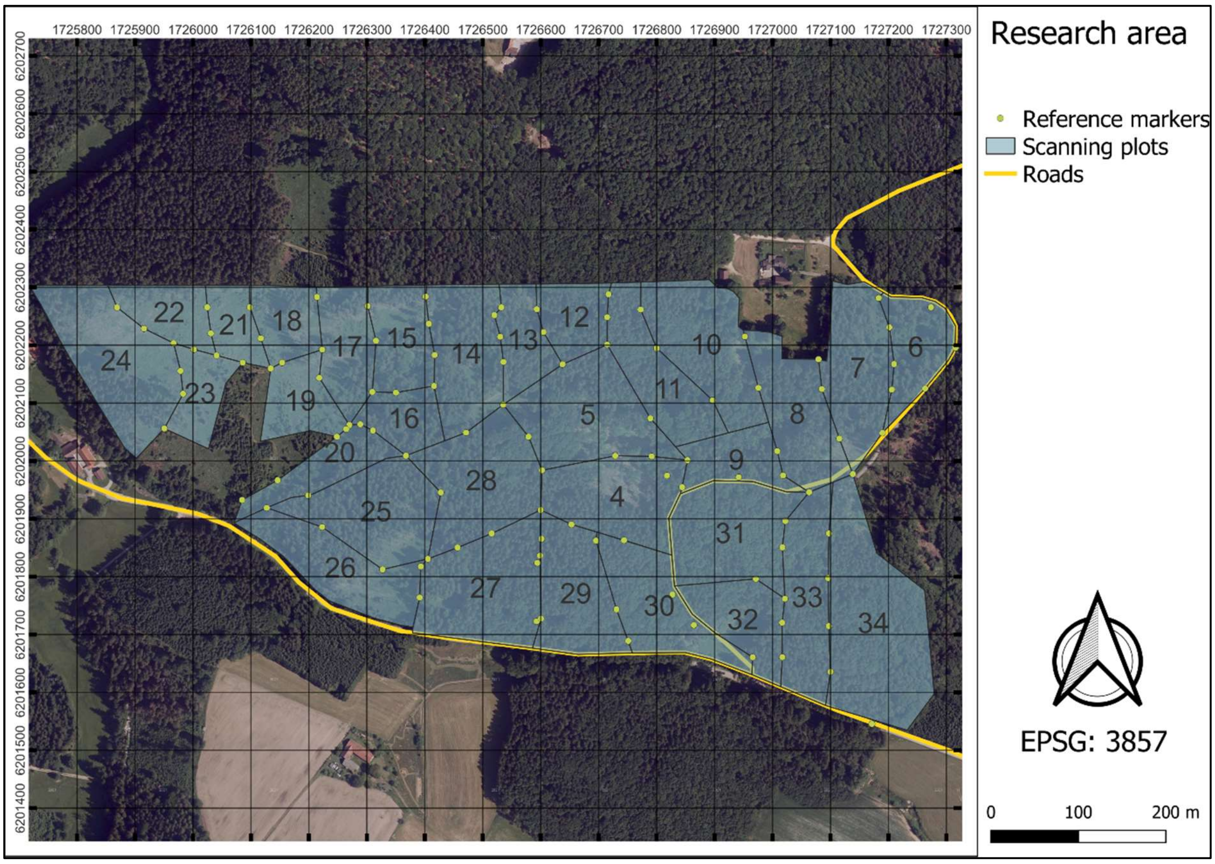


Figure 2: Map of the scanned areas

2.2. Hard- and Software

Processing of the point clouds using SLAM (Simultaneous Localization and Mapping) algorithm was conducted using GeoSLAM Hub, a processing platform provided by the company which produces the scanner. The software allowed for an easy transition of the raw data obtained from the scanner in ".geoslam" – format to point cloud data in ".las" – format. Afterwards, the data could be imported into the workspace of the R software (version 4.2.1), enabling further processing of the data via the statistical programming language R [18]. Since the tree positions are needed as input data for the registration algorithm, an automatic tree detection algorithm, as described by Gollob et al. [19], was applied.

The computations for the registration algorithm were performed using a PC workstation equipped with an Intel® Xeon® W-3223 processor possessing 8 cores and 16 threads. 256 GB of internal memory were available, so that the partly CPU- and memory-intensive computations could be performed without problems.

2.3. Methodology

The approach adapted as part of this thesis was developed and described by Hyyppä et al. [13], who co-registered terrestrial and airborne point clouds by matching the priorly detected tree positions of the two data sets and thus finding a 2D-transformation for the tree locations extracted from the terrestrial point cloud. The algorithm computes a set of feature descriptors for each tree location in both data sets by calculating the distances and angles to the nearest neighbouring trees and finds matching tree pairs by minimizing the Euclidean distance between these feature descriptors. In the course of this study, the described algorithm was adapted in such a way as to enable a co-registration of tree locations from multiple terrestrial point clouds obtained with a Personal Laser Scanner (PLS) and when overlapping areas are small.

2.3.1. Initial algorithm

The purpose of the algorithm developed by Hyyppä et al. [13] was to find the 2D Euclidean transformation between the detected tree locations derived from terrestrial and airborne point clouds. After some adaptions, which will be addressed in section 2.3.2, this algorithm performed well also for the co-registration of only sparsely overlapping PLS point clouds. The initial algorithm described by Hyyppä et al. [13] computes a feature descriptor vector for each tree in both data sets, consisting of the angle and distance to the closest neighbouring tree in each of four quadrants. The characteristic direction, which is essential for the definition of the quadrants, is derived from the direction to the closest neighbouring tree. As illustrated in Figure 3 [13], the characteristic direction and the vector perpendicular to it span the four quadrants needed for the following steps: In each of the quadrants, the closest neighbouring tree within a search radius $R = 10$ m is identified.

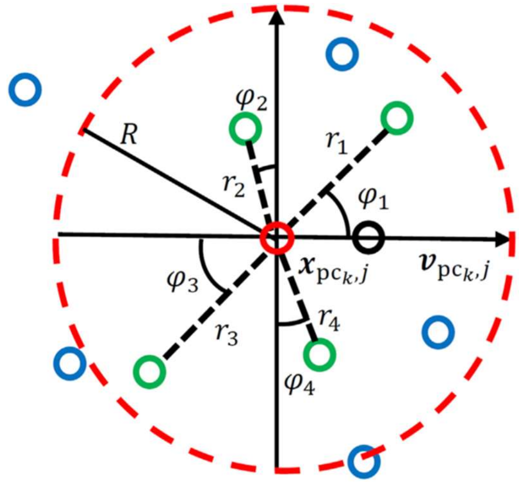


Figure 3: Illustration of the feature descriptors. Adopted from Hyyppä et al. [13]. The quadrants are spanned using the characteristic direction $v_{pc_k,j}$. For each quadrant, the distance (r_{1-4}) and angle (φ_{1-4}) to the closest neighbouring tree within a search radius R is calculated.

The distances and angles from the investigated tree to these closest neighbouring trees constitute the feature descriptors for the eight-digit feature descriptor vector. If no tree is detected within the predefined search radius of 10 m, the corresponding feature descriptors are set to -1. The resulting feature descriptor vectors are the basis for a comparison and thus matching of the point patterns derived from two separate laser scans. Since every tree in each of the data sets has its own feature descriptor vector, it is possible to identify matching trees which were detected in both point clouds by calculating the Euclidean distance between the feature descriptors of trees in data set 1 and the ones in data set 2 [13].

After computing the Euclidean distances, the tree pairs are ranked based on the "2nd nearest neighbour distance ratio" (NNDR), which is defined as follows:

$$NNDR_j = \frac{\left\| f_{pc_2,j} - f_{pc_1,NN(j)} \right\|_2}{\left\| f_{pc_2,j} - f_{pc_1,2nd\quad(j)} \right\|_2} \tag{1}$$

$f_{pc_1,NN(j)}$ is the nearest neighbour descriptor in point cloud 1 (pc_1) to each feature descriptor in point cloud 2 ($f_{pc_2,j}$), whereas $f_{pc_1,2ndNN(j)}$ is the second nearest neighbour descriptor. The smaller the described ratio, the more reliable the tentative matches are. The $k = 20$ most reliable matches, thus the ones with the lowest NNDRs, are afterwards taken to compute the corresponding Euclidean transformations. This approach results in 20 different transformations, which are subsequently applied to both point patterns. After separately shifting and rotating the coordinates according to these 20 transformations, the one with the highest number of matching tree pairs with a Euclidean distance below $r_{thres} = 1$ m is selected as the best fitting transformation.

On a final note, the rotational (θ) and translational (t) parameters of this transformation are taken as initial values for the loss function

$$L(\theta, t) = \sum_j \left\| R(\theta) x_{pc_1, MATCH(j)} + t - x_{pc_2, j} \right\|_2^2 \qquad (2)$$

which is finally minimized by applying an iterative optimization algorithm. The loss function computes the sum of the squared Euclidean distances between the transformed matching pairs in the first data set and the matching pairs in the second data set, where θ and t are rotation and translation, thus the parameters which are being refined. Their initial values are taken from the best Euclidean transformation parameters, which have been identified as described above [13].

Using the described approach, Hyyppä et al. [13] were able to efficiently match terrestrial and airborne point clouds. The authors learned that the proposed algorithm is robust even when the scanned areas greatly differ in size or show only partial overlap, which should make it perfectly suitable as basis for the co-registration of two adjacent terrestrial point clouds and hence for achieving the objectives of this thesis. Moreover, this coarse registration can serve as basis for a finer registration, using for example point-based methods such as ICP (Iterative Closest Point). Since ICP operates by minimizing the distances between the closest neighbouring points in two adjacent scans, this method requires a smaller difference between the initial and the transformed point clouds, and a prior coarse registration - as elaborated in this thesis – is prerequisite for a successful application of ICP [17,20]. The fine registration using the ICP method could be implemented using CloudCompare software [21], which includes a tool for the automatic registration via ICP [22].

2.3.2. Adaptions of the algorithm

2.3.2.1. Parameter settings

In the initial algorithm described by Hyyppä et al. [13], there are essentially three parameters that can be changed: the search radius R, the threshold for the Euclidean distance r_{thre} and the number of matching pairs k, taken for the computation of the k tentative Euclidean transformations. Initial trials have failed to match the tree positions from the laser scans according to descriptions in section 2.1. As solution, caliper thresholds were introduced to minimize the number of misleading points. As depicted exemplarily for Scan 12 and 13 in Figure 4, the filtering of trees with a DBH < 20 cm already led to a successful co-registration of these two scans. However, the optimal caliper threshold is different for other scans and strongly dependent on stand density and number of trees, making it hardly predictable.

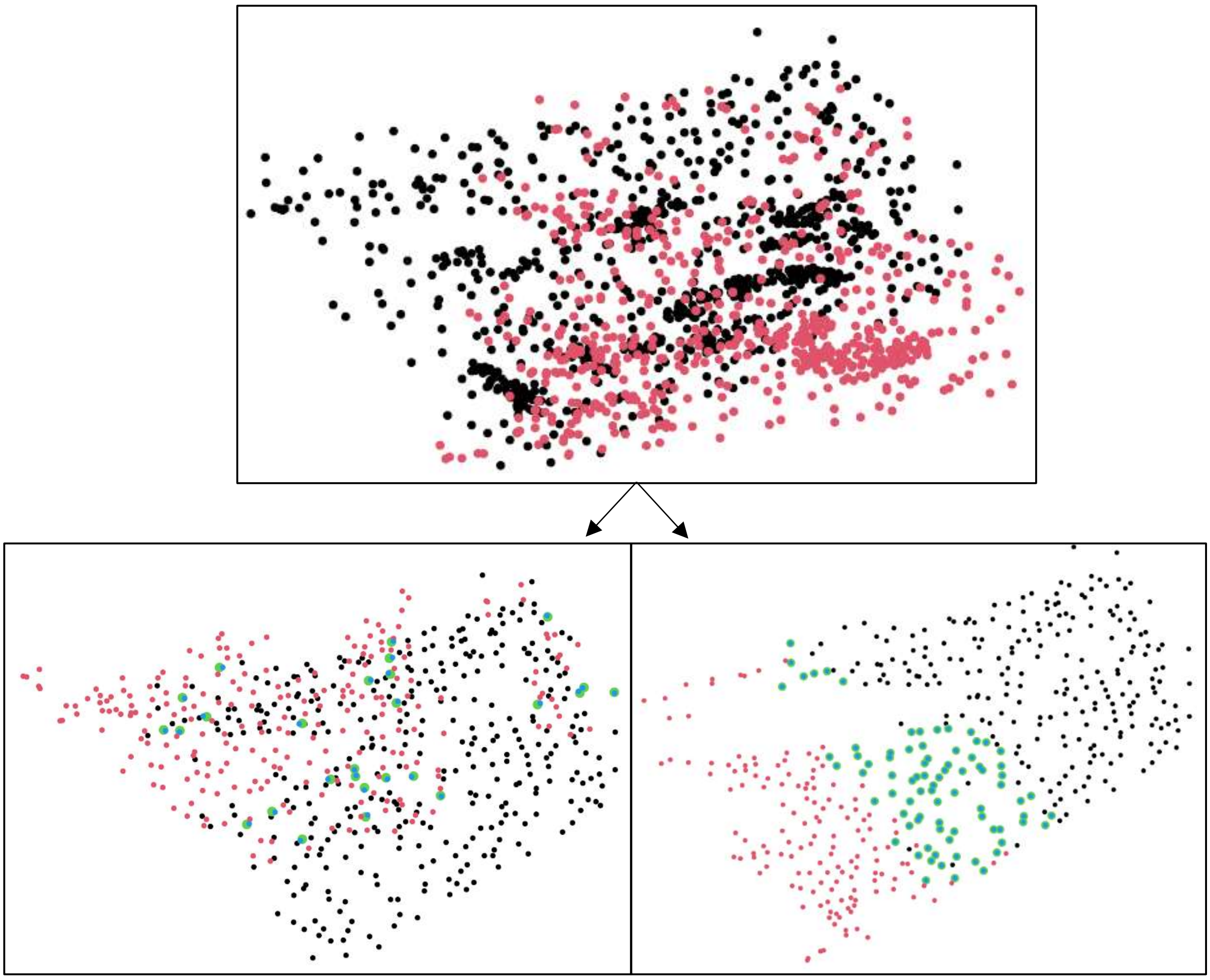


Figure 4: Tree locations of Scan 12 (in black) and Scan 13 (in red). The first picture shows the unregistered tree locations, whereas the second and third picture show the tree locations after the transformation. The green and blue points signify matching tree pairs in Scan 12 and 13 respectively. The right point set results from a co-registration with a prior threshold for the tree diameters set to DBH > 20 cm. The point pattern in the left panel results from a co-registration without any threshold.

A more easily implementable way of optimizing the algorithm for our purposes was to change the parameter k. As mentioned before, this parameter determines the number of tentative matching pairs taken for the computation of k transformations, from which the best one is afterwards identified by comparing the number of matching pairs. A static parameter setting with k = 20 turned out to be unsuitable for most of the scans, leading to inadequate registration results. The algorithm was thus extended in such a way that the optimal number of tentative matching pairs is evaluated beforehand. To illustrate the issue, the transformations derived from k = 100 matching pairs were computed at first. Figure 5 exemplarily shows the resulting x- and y-translations for Scan 13 when matching it with Scan 12, exhibiting congruent and thus appropriate results for the first 6 transformations. The same conclusions can be drawn when examining the rotation parameters. The almost compliant x-translations of the 6 matching pairs with the lowest NNDRs indicate the correctness of the calculated transformations, whereas the other pairs exhibit strongly varying translations.

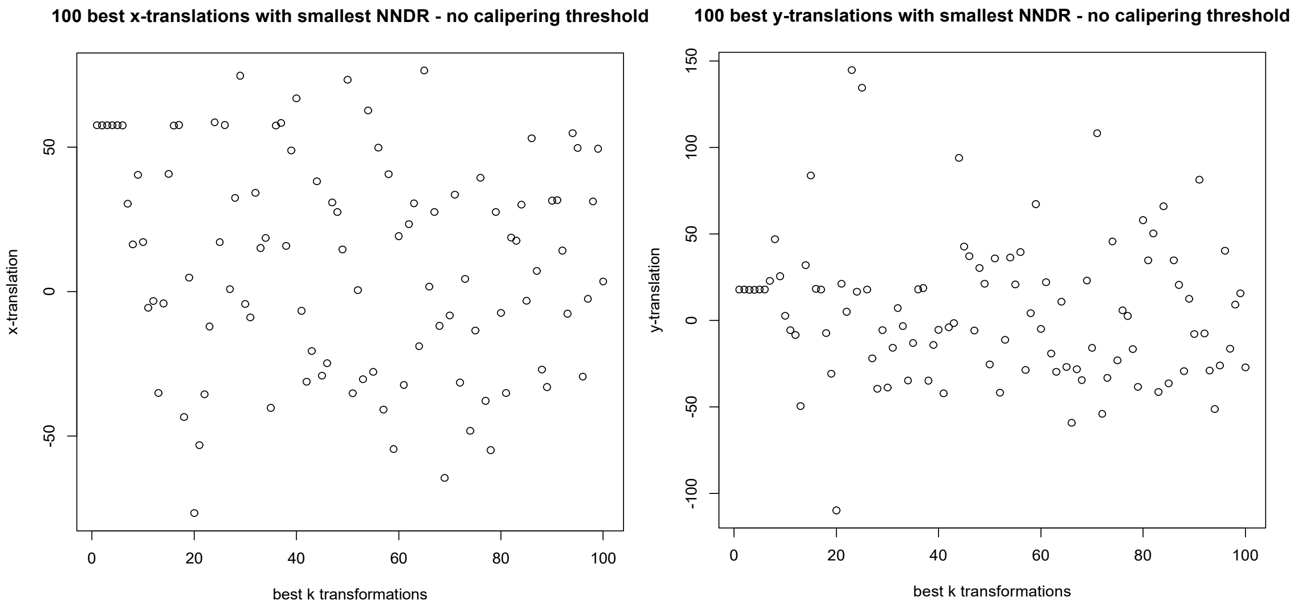


Figure 5: 100 best x- and y-translations when matching Scan 13 to Scan 12 (no caliper threshold)

An examination of this graphic for the same point pattern after filtering all trees with a DBH below 20 cm indicates why the application of this caliper threshold led to good registration results (Fig. 6), since the transformations were congruent for the first 31 tentative matching pairs. Accordingly, without the pre-filtering of points, choosing k = 20 matching pairs for the computation of tentative transformations more likely leads to the selection of wrong transformation parameters.

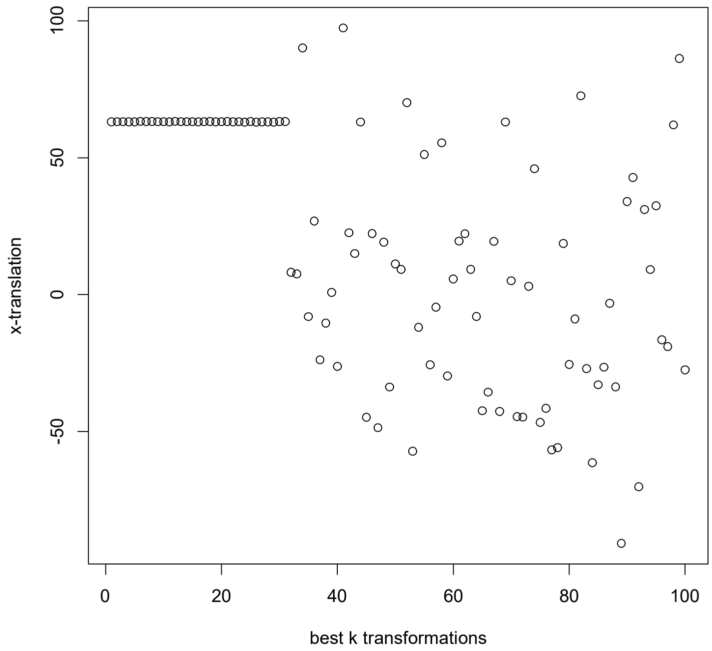


Figure 6: 100 best x- and y-translations when matching Scan 13 to Scan 12 (caliper threshold: 20 cm)

To obtain optimal registration results, the variance between the y-translations (x-translation and rotation would lead to the same results) of the first 60 tentative matching pairs was calculated in steps of 3, i.e., for matches 1-3, for matches 2-4 etc. The value 3 was chosen for this step because 3 transformations were at least necessary to indicate correctness when congruent. Choosing a larger number might have led to false results, as there might only be 3 congruent y-translations for some point patterns. The medial matching pair of the triplet with the smallest variance was finally selected to function as the new parameter setting for k. Using the example of Scans 12 and 13, the hereby selected parameter setting of k = 3 resulted in an adequate co-registration, even without the preliminary thinning of the point pattern through the application of a caliper threshold (Fig. 7).

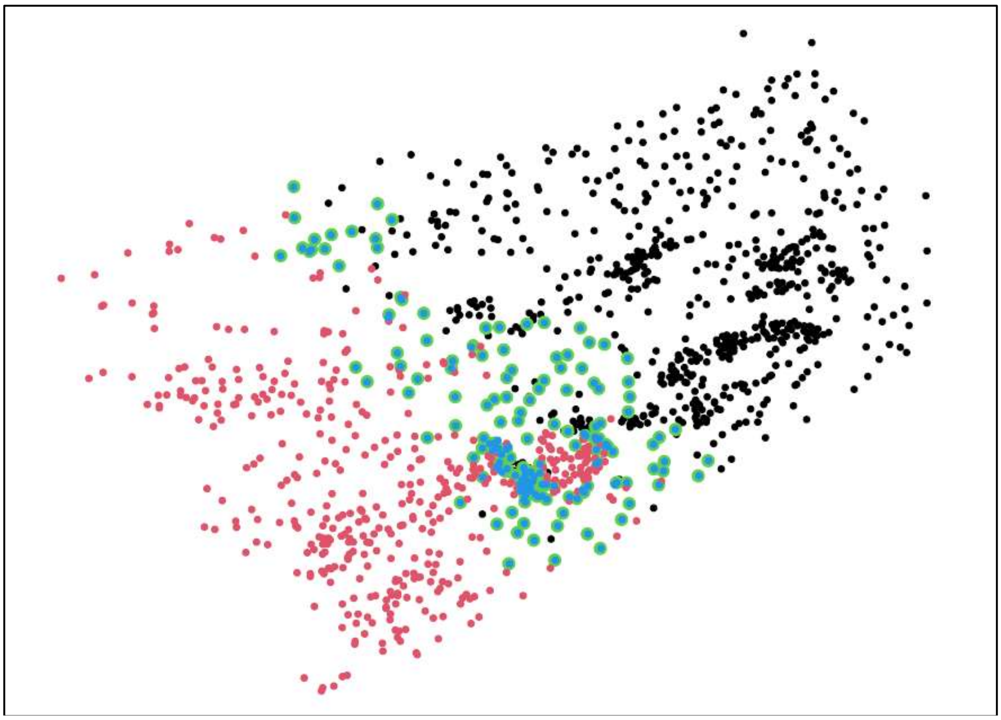


Figure 7: Tree locations of Scan 12 (in black) and Scan 13 (in red). The green and blue points signify matching tree pairs in Scan 12 and 13 respectively. No caliper threshold was applied, the value for parameter k was set to 3.

2.3.2.2. *Registration of multiple scans*

Unlike the task solved by Hyyppä et al. [13], the challenge which was met in the course of this study was the co-registration of a large number of adjacent scans. In order to obtain appropriate transformation parameters for each single scan and to produce a single comprehensive pattern of tree locations, two methods were applied and evaluated.

The first method, which is henceforth referred to as sequential method, was based on the consecutive co-registration of point patterns, which functions as follows: The registration process starts at one point pattern, with the first step being the calculation of the transformation parameters for the second point pattern to fit to this first one. After obtaining the transformed x- and y-coordinates of the second point pattern, the third point pattern is likewise attached to these already transformed coordinates. Thus, there is always one set of tree locations obtained from one scan, which are co-registered referring to the very first scan, and one set of tree locations that has to be matched to it. The second method, applying a cumulative co-registration, starts in a similar way to the sequential method and is henceforth referred to as cumulative method. The second scan is matched to the first scan and the transformed coordinates of the former are saved for the next step. This time however, the third set of coordinates is not only matched to the coordinates of the second one, but to the combined set of both the first and second scan. The described two methods could themselves be varied by selecting different starting scans, which will be further addressed in Sections 3.1.1 and 3.2.1.

3. Results

3.1. Application of the algorithm using sequential co-registration

3.1.1. Tree pair matching

When applying the modified algorithm proposed by Hyyppä et al. [13] with a varying parameter *k* and consecutively matching the coordinate sets of located trees starting from Scan 34 (compare Fig. 1), a contiguous point pattern as depicted in Figure 8 could be obtained. The tree locations seem fitting, roads and open spaces are clearly distinct from forested area.

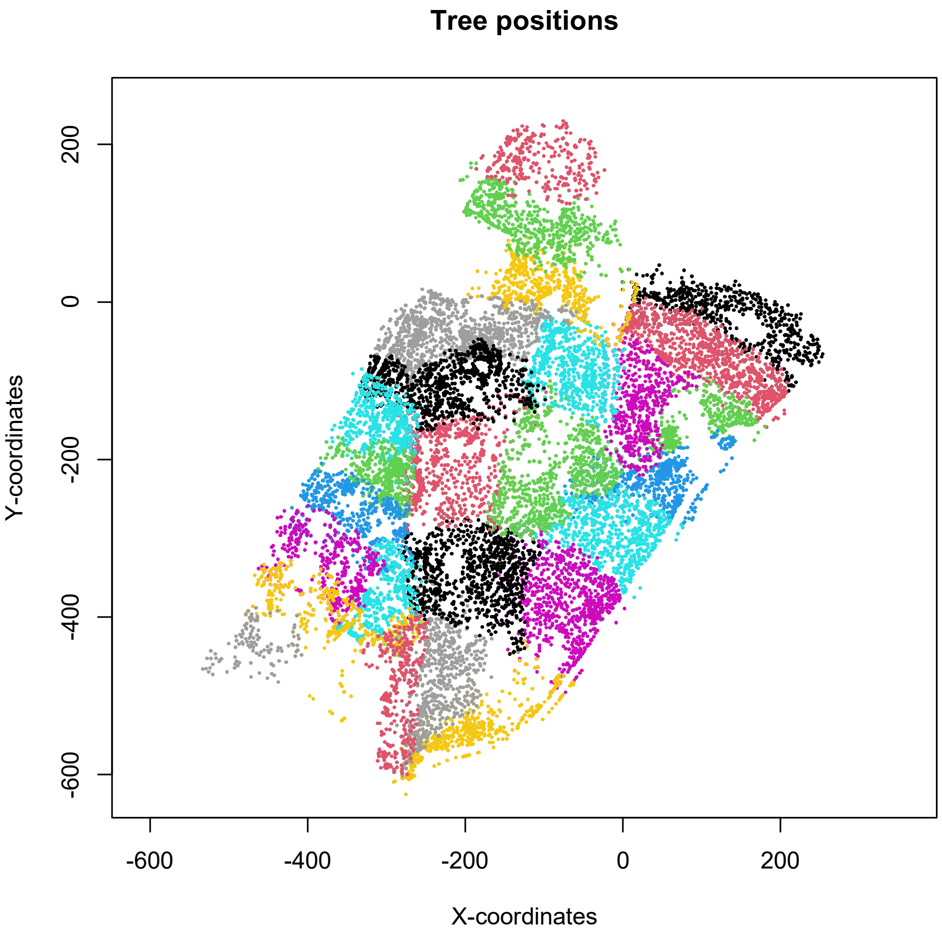


Figure 8: Tree locations after sequential transformation

As can be derived from a closer look at Figure 8, the remaining offset after the application of the above-mentioned method was quite large, especially for point patterns which were adjacent but not directly co-registered. The matching tree locations of the Scans 11 and 5 for example (Fig. 9) displayed a mean deviation of 1.51 m (standard deviation = 0.81 m), with a maximum of up to 3.22 m and a minimum of 0.25 m. Trials with an initial point pattern near the centre of the scanned area were also conducted, since in the semi-automatic manual co-registration using target spheres this approach has turned out to exhibit less error propagation and thus resulted in smaller offsets. However, the deviation between matching tree locations did not considerably change when applying this method, ranging from 0.27 to 3.29 m with a mean of 1.48 m and a standard deviation of 0.93 m.

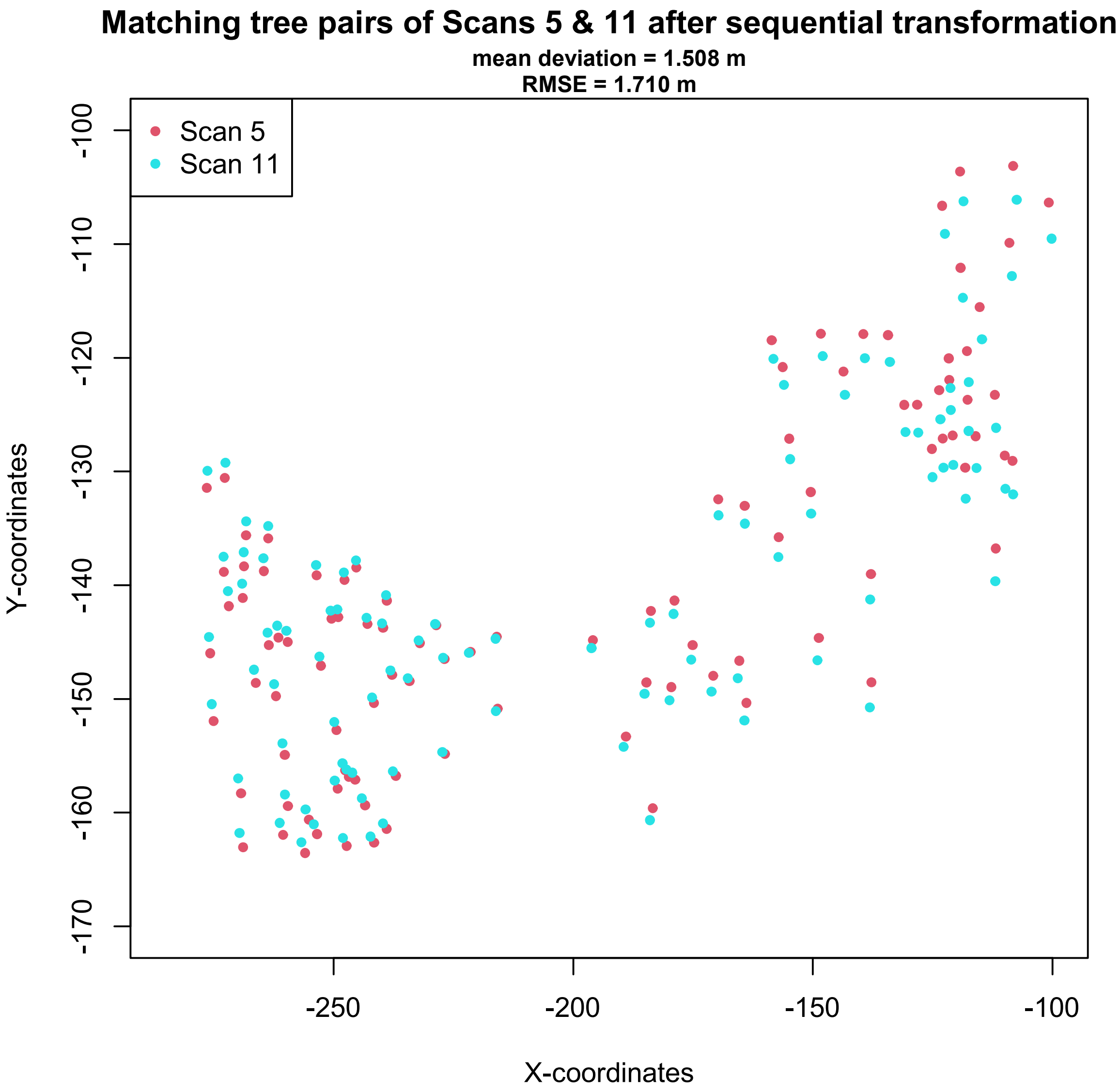


Figure 9: Plotted matching pairs of Scan 5 & 11. The offsets top right of the plot reach over 3 m.

3.1.2. Computation time

The time needed for the computation of all transformations, using the hardware components listed in section 2.2, is depicted in Table 2. Since each point pattern is only matched with one of its directly neighbouring point patterns, the computation time per iteration does not continually increase.

Table 2: Computation time for sequential co-registration

Matched Scans	computation time [min]	Matched Scans	computation time [min]
Scan 34 & 33	2.39	Scan 8 & 10	2.78
Scan 33 & 32	1.85	Scan 10 & 11	3.57
Scan 32 & 30	1.29	Scan 11 & 12	2.09
Scan 30 & 29	1.91	Scan 12 & 13	1.25
Scan 29 & 27	2.00	Scan 13 & 14	0.99
Scan 27 & 26	1.56	Scan 14 & 16	0.91
Scan 26 & 25	1.43	Scan 16 & 15	0.73
Scan 25 & 28	2.07	Scan 15 & 17	0.5
Scan 28 & 5	2.17	Scan 17 & 18	0.28
Scan 5 & 4	1.86	Scan 8 & 7	1.22
Scan 4 & 31	1.27	Scan 7 & 6	0.7
Scan 31 & 9	1.4	Scan 25 & 20	1.09
Scan 9 & 8	1.51		
Total			
26 Scans - c. 30 ha		c. 39 min	

3.2. Application of the algorithm using cumulative co-registration

3.2.1. Tree pair matching

When matching the obtained tree locations cumulatively, again starting at Scan 34, at first appearance a similar output to the one described in section 3.1.1 was obtained (Fig. 10). The remaining offsets between matching pairs of adjacent scans, however, were lower after the application of the cumulative method. To facilitate a closer look at each separate scan after the application of this co-registration method, detailed cuttings of each tree location point pattern with its adjacent point patterns are provided in Appendix A.

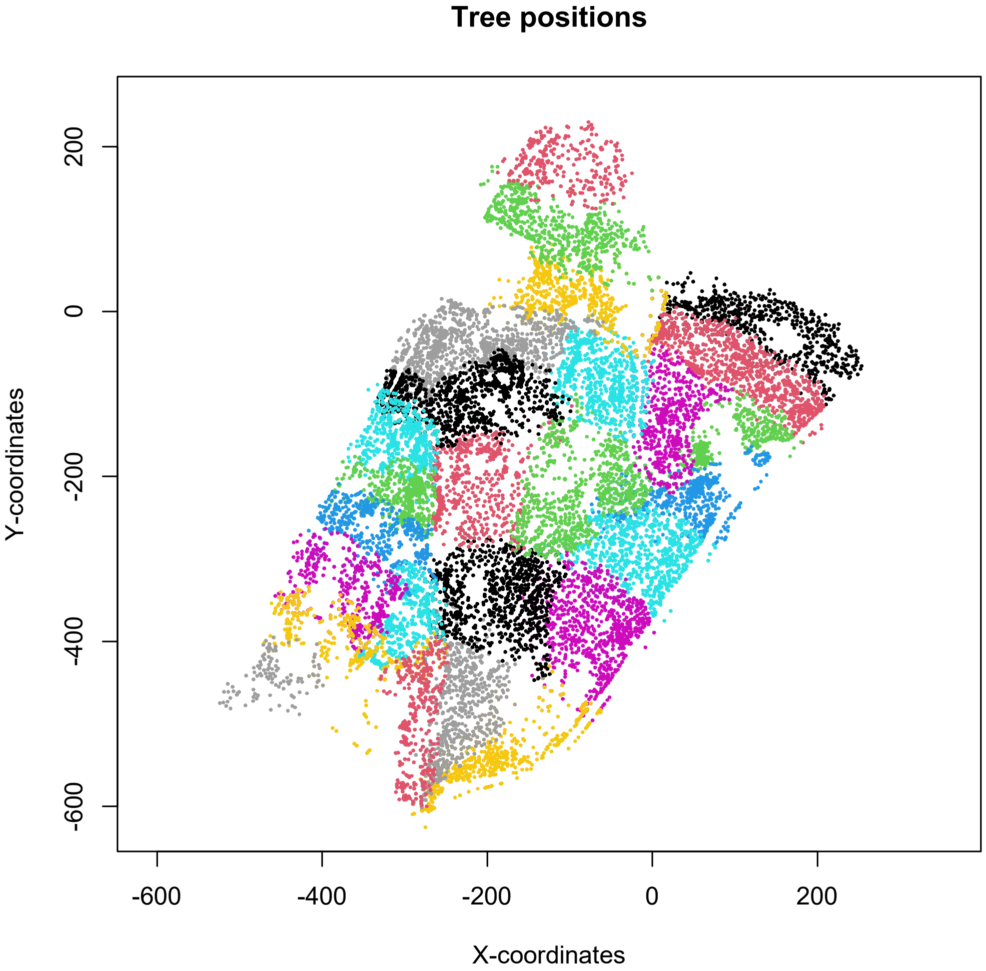


Figure 10: Tree locations after cumulative transformation

Especially when once again taking a closer look at the exemplary clipped section of Scans 11 and 5 (Fig. 11), the difference becomes obvious. The offsets between the same matching points which have been analysed in Section 3.1.1 ranged between 0.44 cm and 19.20 cm, with a mean deviation of 5.10 cm (standard deviation = 3.19 cm). When choosing an initial point pattern near the centre of the scanned area, the deviation between matching tree locations ranged between 0.48 cm and 20.66 cm with a mean of 4.50 cm and a standard deviation of 3.31 cm.

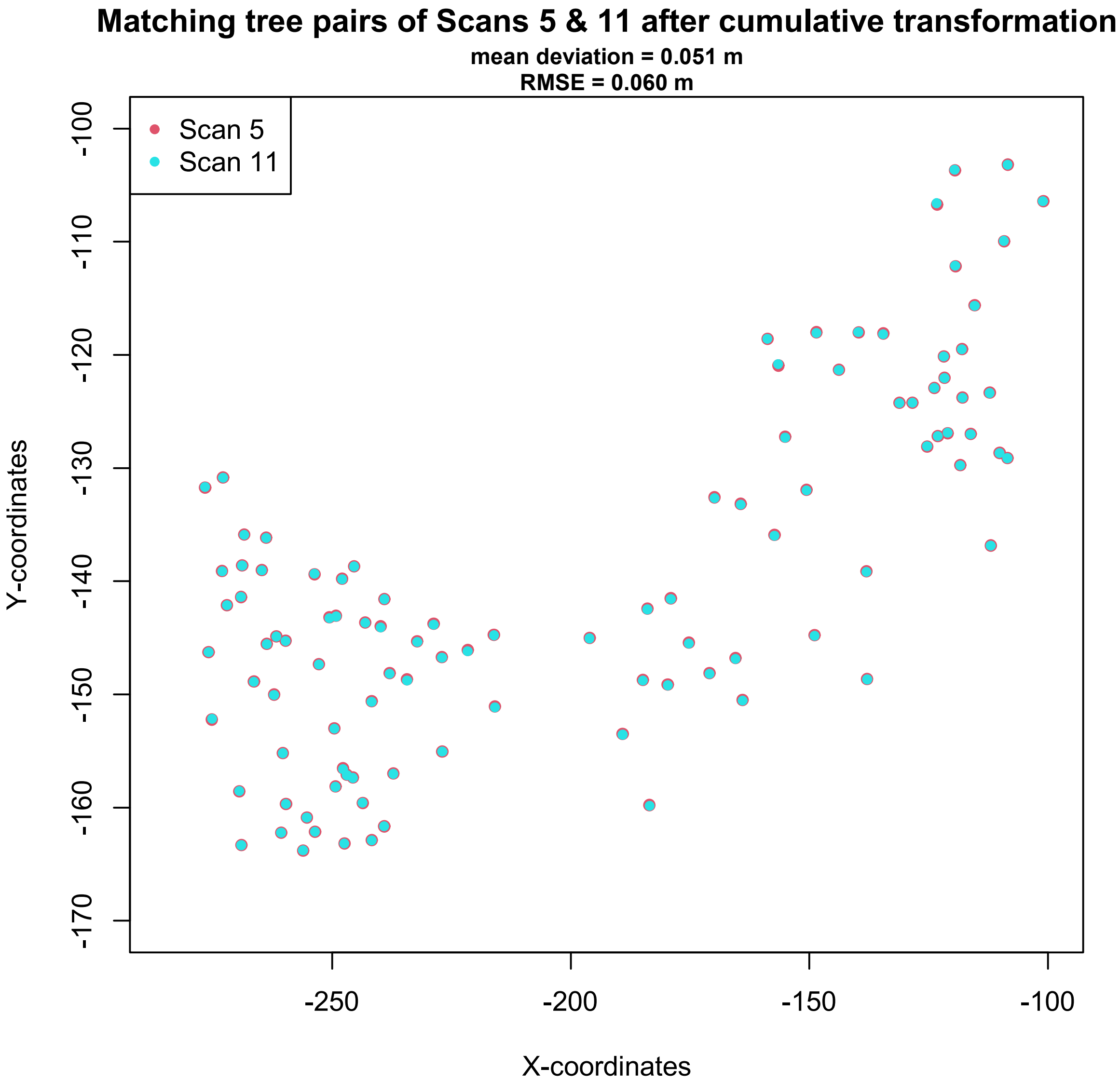


Figure 11: Plotted matching pairs of Scan 5 & 11. The offsets never exceed 20 cm.

3.2.2. Computation time

The time needed for the computation of all transformations, using the hardware components listed in section 2.2, is depicted in Table 3. Since the number of points to which a point pattern is matched rises with each iteration, the computation time per iteration increases accordingly. Scan 10 had to be registered separately because of registration errors. They probably occurred due to a corrupted point cloud and tree location errors. This would also explain the relatively large offsets of Scan 10, which are clearly recognizable in Appendix A, even after the direct registration only with its neighbours.

Table 3: Computation time for cumulative co-registration

Matched Scans	computation time [min]	Matched Scans	computation time [min]
Scan 34 & 33	2.36	previous &10	204.97
previous & 32	5.04	previous & 11	234.06
previous & 30	9.21	previous & 12	281.95
previous & 29	16.39	previous & 13	306.56
previous & 27	25.45	previous & 14	331.84
previous & 26	36.34	previous & 16	353.72
previous & 25	47.97	previous & 15	407.99
previous & 28	65.61	previous & 17	429.27
previous & 5	87.29	previous & 18	446.46
previous & 4	106.94	previous & 20	519.80
previous & 31	125.23	previous & 7	493.11
previous & 9	140.07	previous & 6	512.67
previous & 8	177.78	Scan 10 and adjacent	11.99
Total			
26 Scans - c. 30 ha		c. 89 hours / 3.7 days	

4. Discussion

4.1. Parameter settings

The parameter which was varied to optimize the registration results was k, the number of matching pairs selected for the computation of tentative transformation parameters. As described in section 2.3.2.1, these k transformations were applied to all tree locations of one point pattern to identify the transformation with the highest number of matching trees. Hyyppä et al. [13] suggest that suitable values for the parameter k lie between 10 and 50, depending on stem density and estimation errors of the tree locations [13]. Since these variables change with each scan, an adaptable parameter k seems to be a reasonable solution. As mentioned before, the temporary number of k, analysed for the identification of the optimal value for this parameter, was 60. The selection of this value allowed for a reliable determination of the optimal value for the parameter k, as the range suggested by Hyyppä et al. [13] is covered with certainty.

4.2. Sequential versus cumulative co-registration

The crucial difference between the two conducted methods for the registration of multiple adjacent point patterns is the accuracy of the transformations. As mentioned in the previous sections, the accuracies which can be achieved with sequential co-registration contrast strongly with those reachable when cumulatively matching the point patterns. Since the obtainment of one contiguous point cloud for the whole area could be useful for a variety of applications, a fine registration using point-based methods such as PCA should be possible, only functioning with already quite closely matched point clouds [17,20]. Thus, a high registration accuracy is a fundamental requirement of the algorithm. As a result, a distance between two matching trees of over 3 m, as was sometimes the case for the sequential registration method, disqualifies the latter for all further applications.

The great registration errors of the mentioned method are probably a consequence of error propagation [23]. As outlined by Evangelidis et al. [24], these “sequential register-then-integrate strategies” only lead to optimal results when applied locally, otherwise falling victim to error propagation. This assumption could explain why adjacent, yet not directly co-registered point patterns displayed large offsets while point patterns directly matched with each other did not show these inaccuracies. The sequential method can be considered a pairwise registration method, contrary to global methods, meaning that transformation parameters are always computed between two point sets. Since point correspondences might not only exist between the co-registered points when operating on multiple point sets, global methods are considered to be a better alternative for this case of application [25]. The cumulative approach applied in the course of this thesis resembles these global, also known as group wise, registrations to the extent that it considers all adjacent point sets which might have correspondences with the not yet transformed points [23]. The high accuracy of the applied method results from the fact that, in the end, each point set is finally correlated with all its neighbouring point sets, leaving less room for potential errors.

Another downside of the sequential method is the impossibility of performing the point set registration in one run. Depending on the spatial distribution of the conducted scans, there were always one or more point patterns which could not be integrated into the "registration sequence" because, doing so, the algorithm was stuck in a false status beyond the global optimum. Thus, some point patterns had to be added separately, implying additional work and losses in efficiency. The cumulative method, however, allowed for the registration of each point set regardless of any predefined sequence, the only conditions being that the new set of points is adjacent to the already registered accumulation of point sets and that it contains enough trees which overlap with the latter. Scan 19 did not fulfil the second condition, as the scanned area was a deforested area with only few, scattered trees spread over it. The overlapping areas need to contain a high number of trees for the registration algorithm to work optimally, which was not sufficiently considered during the scanning process and thus lead to the exclusion of scans 19, 21, 22, 23 and 24.

The drawback of the more accurate, cumulative registration method is its high computation time, which increases with each iteration, such as already mentioned in Section 3.2.2. Since the feature descriptor vectors for every single point in each of the point sets as well as the Euclidean distances between them must be calculated, computation time inevitably rises with increasing point number, as is the case when cumulatively matching the point sets. Undoubtedly, the original MATLAB version of the registration algorithm developed by Hyyppä et al. [13] would perform much faster than its modified R version, as R is known to be the slower of these two programming languages [26]. However, the computation time seems tolerable, especially when the relatively large extend of the study area is considered.

4.3. Outlook

Since the starting positions of multiple scans are usually not exactly in the same height and each scan produces its own local coordinate system, the final step to complete the coarse registration would be the transformation of the z-coordinate. Wang et al. [7] for example computed the vertical translation by calculating the differences between the z-coordinates of matching tree pairs and averaging the results over the total number of matches, which is possible because modern scanners usually produce well-levelled point clouds. This approach implies that each matching tree position in one point pattern can be assigned to its corresponding position in the other point pattern. An easily implementable way of doing this, which has already been tested successfully, is to identify trees which are closer than a certain threshold by applying the function “connected.ppp” from the spatstat.geom package [27]. Tree pairs which are closer than the chosen threshold and exhibit different scan IDs can be regarded as matching pairs. Taking the mean of their respective attributes, such as DBH, height or z-coordinate, allows for the merging of trees which would otherwise be duplicated and thus results in an accurate list of all scanned trees. From this list, stand parameters such as stem density or volume of standing timber can be calculated. As already mentioned, the knowledge of corresponding tree pairs also allows for the transformation of the z-coordinate following the approach described by Wang et al. [7].

After the completion of this step, a fine registration based on the coarsely transformed point clouds can be performed. The open-source program CloudCompare [21] includes a tool for the fine registration of point clouds based on an ICP algorithm, which could offer a solution for this final task of improvement. The input data required for this algorithm are basically two point clouds: a “data” point cloud, which should be transformed and a “model”-point cloud, which acts as reference. For each point of the data point cloud, the ICP algorithm in CloudCompare calculates the closest point in the model point cloud and afterwards minimizes the RMSD (Root mean squared deviation) between these points by applying iteratively optimized transformation parameters on the data point cloud. Adjustable parameters for this process include the number of iterations and a scaling parameter [28]. Although Rajendra et al. [20] pointed out that other ICP algorithms, such as Brute Force or KDTree, perform better than the algorithm implemented in CloudCompare when it comes to speed and accuracy, the latter still produced promising results in various studies [22,29].

According to Hyyppä et al. [13], there are several possibilities for improving the efficiency and accuracy of the coarse registration algorithm. If desired, additional parameters can be added to the feature descriptor vectors, such as DBH or height of the trees. The computation of these feature descriptors could be parallelized in some degree and the identification of the nearest neighbouring trees can be accelerated by space-partitioning [13]. For the size of the given research area however, computation time kept within reasonable bounds and thus no further adjustments to increase time efficiency were made to the algorithm. The inclusion of the DBH as additional parameter was not considered as necessary, since the described registration method led to adequate results and every additional parameter would only lead to higher computation times.

5. Summary and Conclusions

The efficient co-registration of point clouds obtained from Personal Laserscanning is an indispensable component of modern forest inventory based on LiDAR technology. The time- and labour-consuming target-based registration has long been a major obstacle, contributing eminently to the scepticism of many forestry companies towards the usage of PLS for forest inventory purposes. Therefore, alternative methods for an efficient co-registration need to be evaluated in order to put LiDAR technology into practice.

In line with this, the goal of this thesis was to assess an algorithm in terms of its utility for the target-free coarse registration of PLS-point clouds and, if necessary, make adaptions to adjust to the given circumstances and tasks. To obtain practically relevant data for the testing of the algorithm, 30 ha of forested area were scanned in 26 scans using the Personal Laser Scanner (PLS) GeoSLAM ZEB Horizon (GeoSLAM Ltd., Nottingham, UK). The selected algorithm, described by Hyyppä et al. [13], is based on the spatial distribution of trees and their postitions relative to neighboring trees, making it resistant to any given translation or roatation. Some adjustments were made in order to minimize the input required for the algorithm, leaving only the tree locations and the neighborhoods of each scan as necessary input.

The resulting transformations exhibited adequate accuracy, allowing for a further fine registration of the point clouds themselves. Concerning the array of registered tree location patterns, a cumulative method with each point pattern being matched to the accumulated total of the already registered point patterns led to the best results. A sequential approach however, with each point pattern being matched only to its previously registered, direct neighbor, resulted in large offsets to neighboring points which have not been directly registered.

In summary, the described approach can lay the foundation of an efficient, target-free point cloud registration, enabling the fast and precise inventory of large forested areas without the need to spare time and space for the effortful target-based registration. The resulting redundancy of the latter could ideally pave the way for a broader usage of LiDAR technology in practical forest inventory.

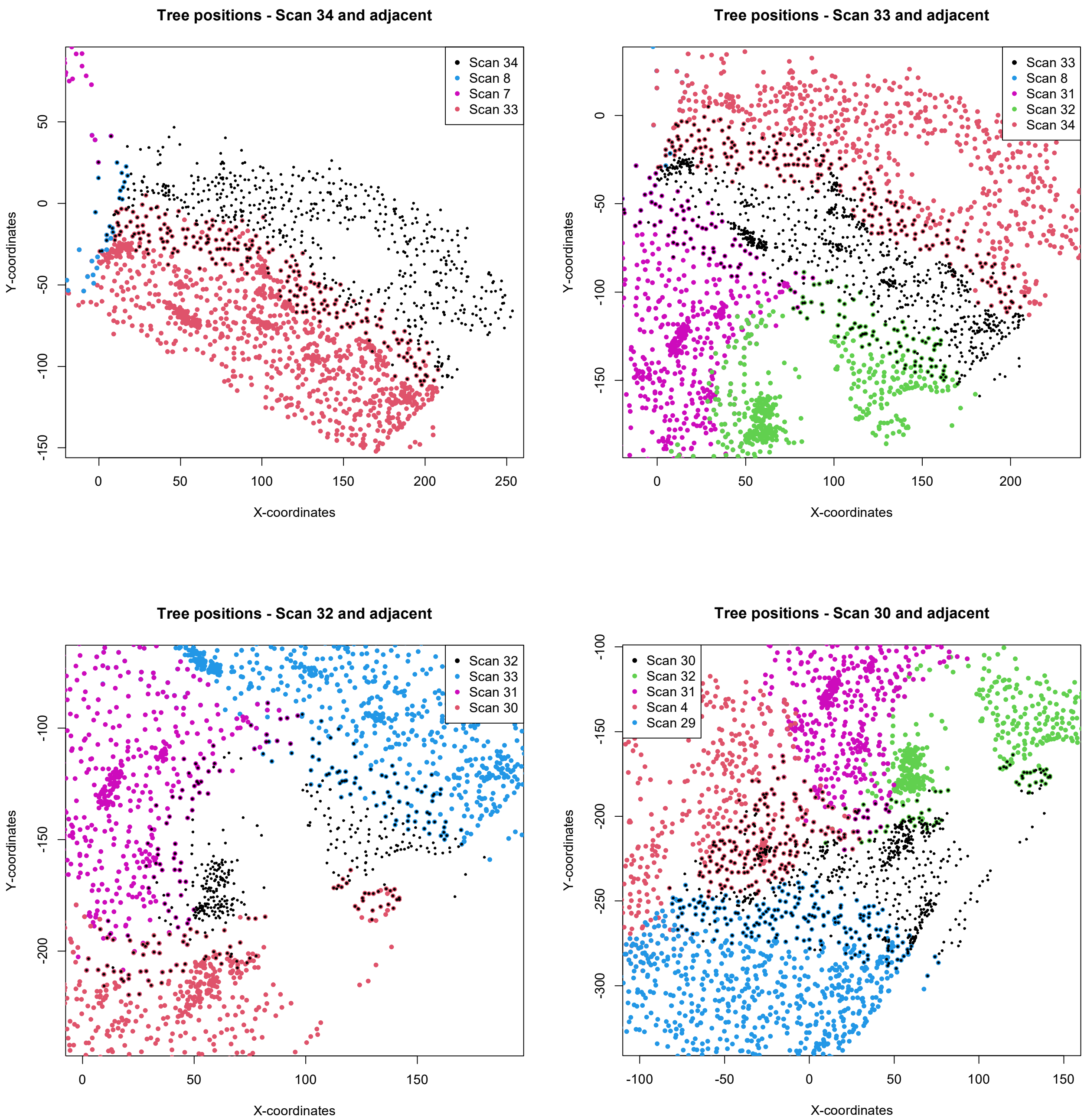
References

1. Graves, H.S. *Forest Mensuration*; John Wiley & Sons: New York, 1906; Vol. First Edition;
2. Liang, X.; Kankare, V.; Hyyppä, J.; Wang, Y.; Kukko, A.; Haggrén, H.; Yu, X.; Kaartinen, H.; Jaakkola, A.; Guan, F.; et al. Terrestrial Laser Scanning in Forest Inventories. *ISPRS Journal of Photogrammetry and Remote Sensing* **2016**, *115*, 63–77, doi:10.1016/J.ISPRSJPRS.2016.01.006.
3. Chen, S.; Liu, H.; Feng, Z.; Shen, C.; Chen, P. Applicability of Personal Laser Scanning in Forestry Inventory. *PLoS One* **2019**, *14*, doi:10.1371/journal.pone.0211392.
4. Balenović, I.; Liang, X.; Jurjević, L.; Hyyppä, J.; Seletković, A.; Kukko, A. Hand-Held Personal Laser Scanning – Current Status and Perspectives for Forest Inventory Application. *Croatian Journal of Forest Engineering* **2020**, *42*, 165–183, doi:10.5552/crojfe.2021.858.
5. Li, J.; Huang, S.; Cui, H.; Ma, Y.; Chen, X. Automatic Point Cloud Registration for Large Outdoor Scenes Using a Priori Semantic Information. *Remote Sens (Basel)* **2021**, *13*, doi:10.3390/rs13173474.
6. Yang, H.; Shi, J.; Carlone, L. TEASER: Fast and Certifiable Point Cloud Registration. **2020**.
7. Wang, X.; Yang, Z.; Cheng, X.; Stoter, J.; Xu, W.; Wu, Z.; Nan, L. Efficient Registration of Forest Point Clouds by Global Matching of Relative Stem Positions. **2021**.
8. Guan, H.; Zhang, J.; Ma, Q.; Liu, M.; Wu, F.; Guo, Q.; Su, Y.; Hu, T.; Wang, R.; Ma, Q.; et al. A Novel Framework to Automatically Fuse Multiplatform LiDAR Data in Forest Environments Based on Tree Locations. *IEEE Transactions on Geoscience and Remote Sensing* **2020**, *58*, 2165–2177, doi:10.1109/TGRS.2019.2953654.
9. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. *PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation*;
10. Johnson, A.E. *Spin-Images: A Representation for 3-D Surface Matching*; 1997;
11. Klein, I.; Filin, S. *LiDAR and INS Fusion in Periods of GPS Outages for Mobile Laser Scanning Mapping Systems*;
12. Cho, H.; Hong, S.; Kim, S.; Park, H.; Park, I.; Sohn, H.G. Application of a Terrestrial LIDAR System for Elevation Mapping in Terra Nova Bay, Antarctica. *Sensors (Switzerland)* **2015**, *15*, 23514–23535, doi:10.3390/s150923514.
13. Hyyppä, E.; Muhojoki, J.; Yu, X.; Kukko, A.; Kaartinen, H.; Hyyppä, J. Efficient Coarse Registration Method Using Translation- and Rotation-Invariant Local Descriptors towards Fully Automated Forest Inventory. *ISPRS Open Journal of Photogrammetry and Remote Sensing* **2021**, *2*, 100007, doi:10.1016/j.ophoto.2021.100007.
14. Polewski, P.; Yao, W.; Cao, L.; Gao, S. Marker-Free Coregistration of UAV and Backpack LiDAR Point Clouds in Forested Areas. *ISPRS Journal of Photogrammetry and Remote Sensing* **2019**, *147*, 307–318, doi:10.1016/j.isprsjprs.2018.11.020.
15. Wendt, A. A Concept for Feature Based Data Registration by Simultaneous Consideration of Laser Scanner Data and Photogrammetric Images. *ISPRS Journal of Photogrammetry and Remote Sensing* **2007**, *62*, 122–134, doi:10.1016/j.isprsjprs.2006.12.001.
16. Guan, H.; Zhang, J.; Ma, Q.; Liu, M.; Wu, F.; Guo, Q.; Su, Y.; Hu, T.; Wang, R.; Ma, Q.; et al. A Novel Framework to Automatically Fuse Multiplatform LiDAR Data in Forest

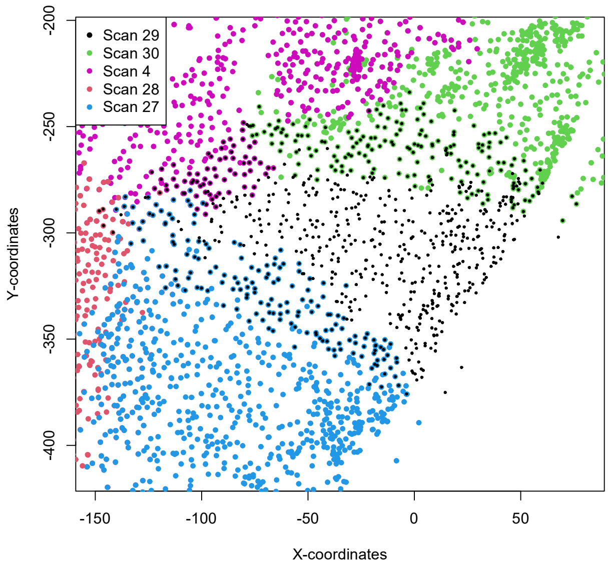
Environments Based on Tree Locations. *IEEE Transactions on Geoscience and Remote Sensing* **2020**, *58*, 2165–2177, doi:10.1109/TGRS.2019.2953654.

17. Besl, P.J.; McKay, N.D. A Method for Registration of 3-D Shapes. *IEEE Trans Pattern Anal Mach Intell* **1992**, *14*, 239–256, doi:10.1109/34.121791.
18. R Core Team R: A Language and Environment for Statistical Computing 2020.
19. Gollob, C.; Ritter, T.; Nothdurft, A. Forest Inventory with Long Range and High-Speed Personal Laser Scanning (PLS) and Simultaneous Localization and Mapping (SLAM) Technology. *Remote Sens (Basel)* **2020**, *12*, doi:10.3390/RS12091509.
20. Rajendra, Y.D.; Mehrotra, S.C.; Kale, K. v.; Manza, R.R.; Dhumal, R.K.; Nagne, A.D.; Vibhute, A.D. Evaluation of Partially Overlapping 3D Point Cloud's Registration by Using ICP Variant and Cloudcompare. In Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives; International Society for Photogrammetry and Remote Sensing, 2014; Vol. XL–8, pp. 891–897.
21. Girardeau-Montaut, D.C. 3D Point Cloud and Mesh Processing Software 2017.
22. Ahmad Fuad, N.; Yusoff, A.R.; Ismail, Z.; Majid, Z. Comparing the Performance of Point Cloud Registration Methods for Landslide Monitoring Using Mobile Laser Scanning Data. In Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives; International Society for Photogrammetry and Remote Sensing, October 26 2018; Vol. 42, pp. 11–21.
23. Zhu, H.; Guo, B.; Zou, K.; Li, Y.; Yuen, K.V.; Mihaylova, L.; Leung, H. A Review of Point Set Registration: From Pairwise Registration to Groupwise Registration. *Sensors (Switzerland)* 2019, *19*.
24. Evangelidis, G.D.; Kounades-Bastian, D.; Horaud, R.; Psarakis, E.Z. A Generative Model for the Joint Registration of Multiple Point Sets. In Proceedings of the Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer Verlag, 2014; Vol. 8695 LNCS, pp. 109–122.
25. Williams, J.; Bennamoun, M. Simultaneous Registration of Multiple Corresponding Point Sets. *Computer Vision and Image Understanding* **2001**, *81*, 117–142, doi:10.1006/cviu.2000.0884.
26. Bora¼ Gan Aruoba, S.; Fernández-Villaverde, J. *A Comparison of Programming Languages in Macroeconomics*; 2015;
27. Baddeley, A.; Turner, R. Spatstat: An R Package for Analyzing Spatial Point Patterns. *J Stat Softw* **2005**, *12*, 1–42.
28. Bytyqi, G. *Diplomarbeit Detektion von Oberflächenveränderungen Mit Terrestrischem Laserscanner Zur Erlangung Des Akademischen Grades Geodäsie Und Geoinformation*;
29. Rahman, F.; Kartini, G.A.J.; Nugroho, H. The Comparison of Point Cloud Registration Quality on Maptek I-Site Studio and Cloud Compare. **2019**.

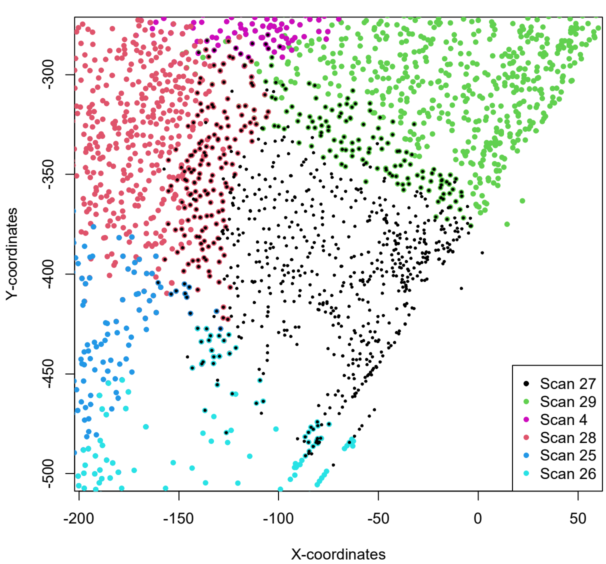
Appendix A


Tree positions - Scan 34 and adjacent
Scan 34
Scan 8
Scan 7
Scan 33
X-coordinates
Y-coordinates
Tree positions - Scan 33 and adjacent
Scan 33
Scan 8
Scan 31
Scan 32
Scan 34
X-coordinates
Y-coordinates
Tree positions - Scan 32 and adjacent
Scan 32
Scan 33
Scan 31
Scan 30
X-coordinates
Y-coordinates
Tree positions - Scan 30 and adjacent
Scan 30
Scan 32
Scan 31
Scan 4
Scan 29
X-coordinates
Y-coordinates

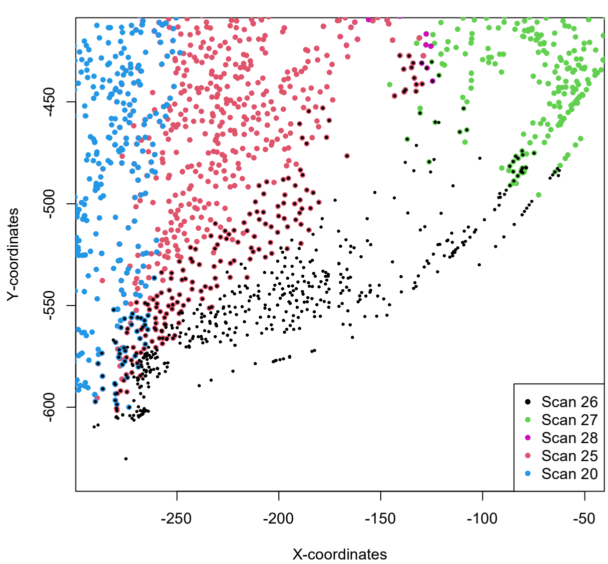
Tree positions - Scan 29 and adjacent



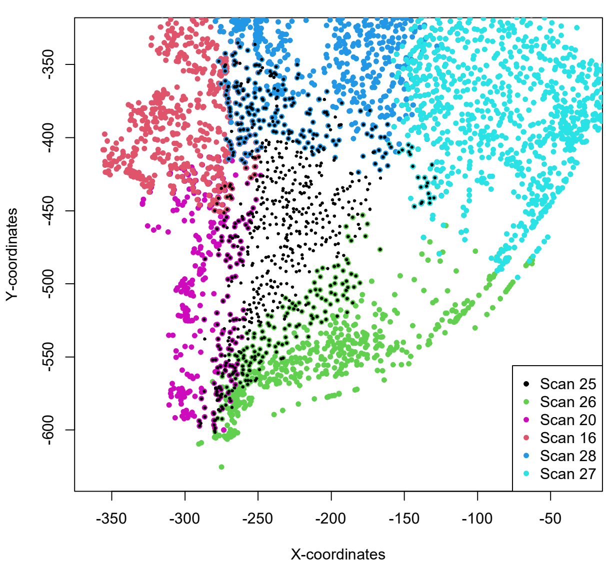
Tree positions - Scan 27 and adjacent



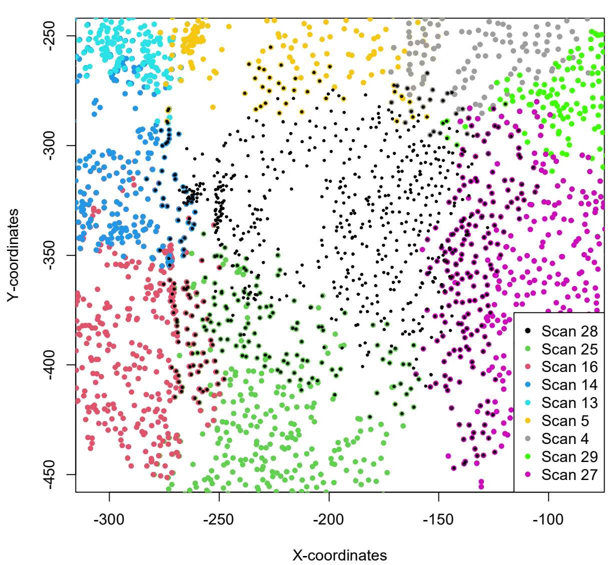
Tree positions - Scan 26 and adjacent



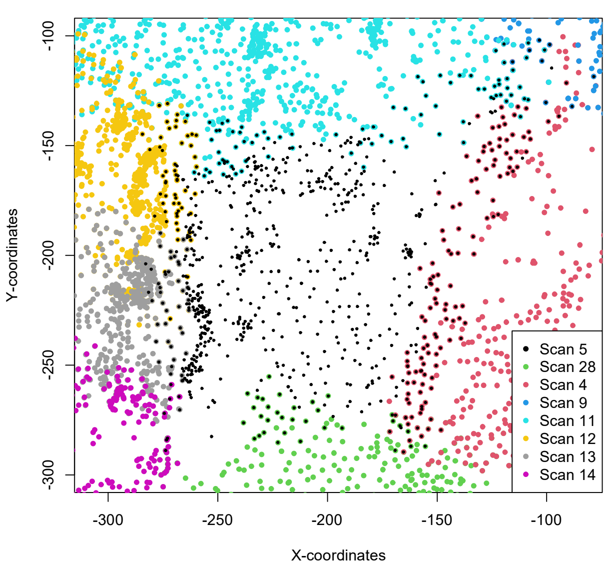
Tree positions - Scan 25 and adjacent



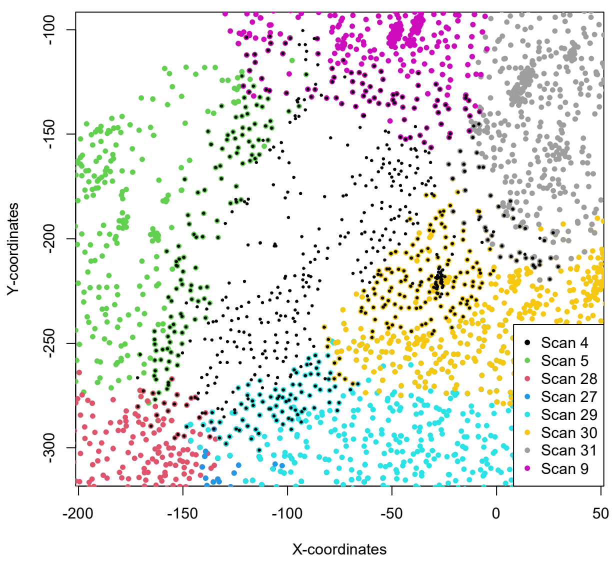
Tree positions - Scan 28 and adjacent



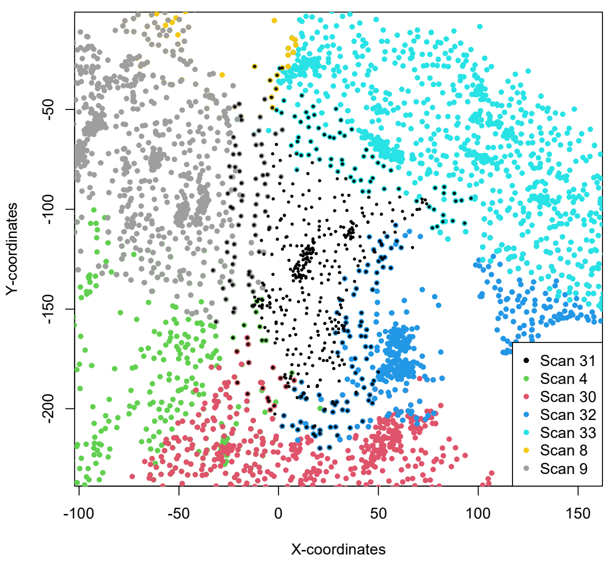
Tree positions - Scan 5 and adjacent



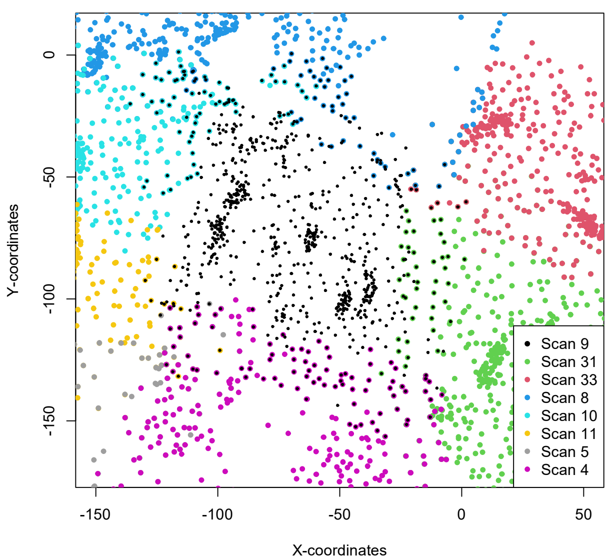
Tree positions - Scan 4 and adjacent



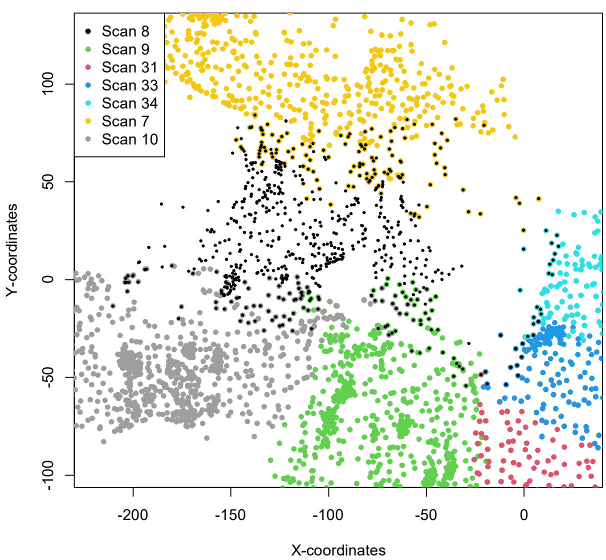
Tree positions - Scan 31 and adjacent



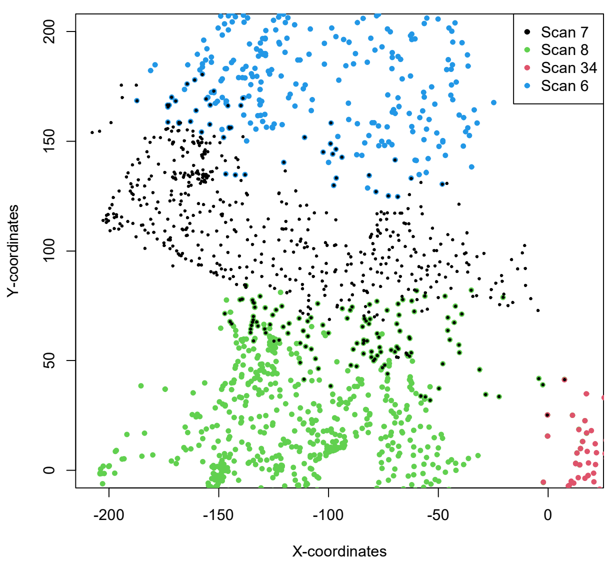
Tree positions - Scan 9 and adjacent



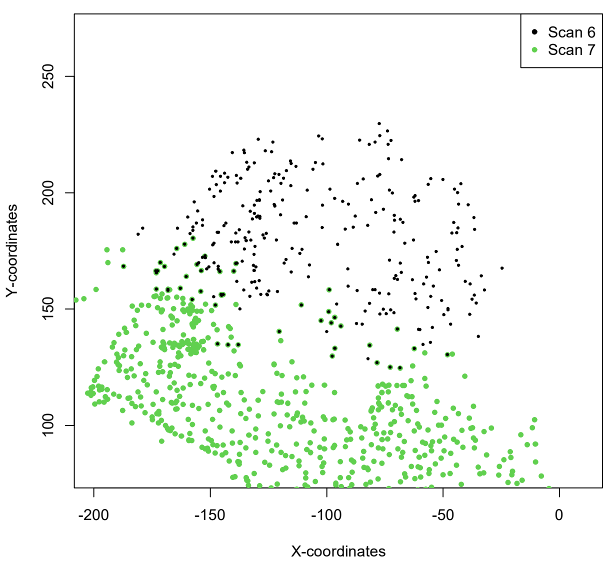
Tree positions - Scan 8 and adjacent

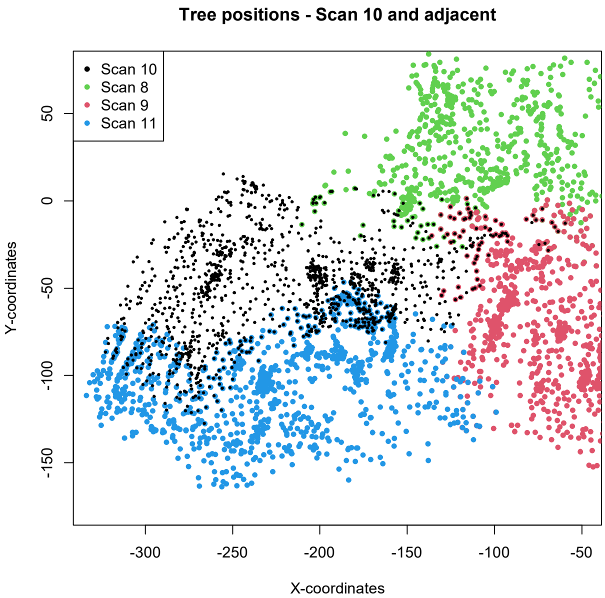


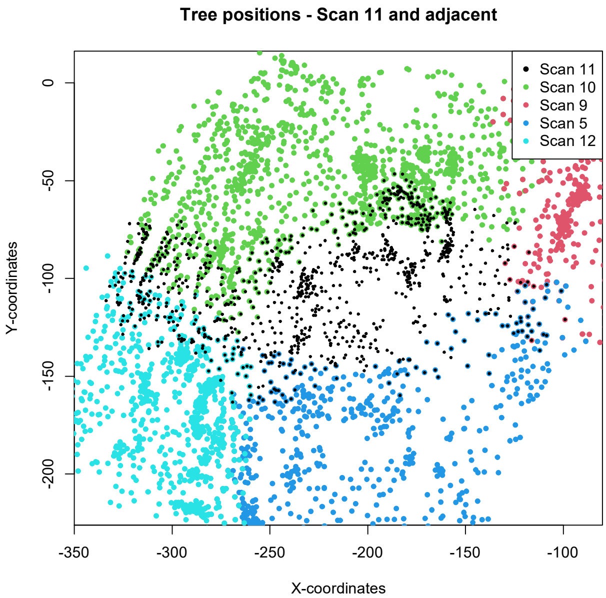
Tree positions - Scan 7 and adjacent

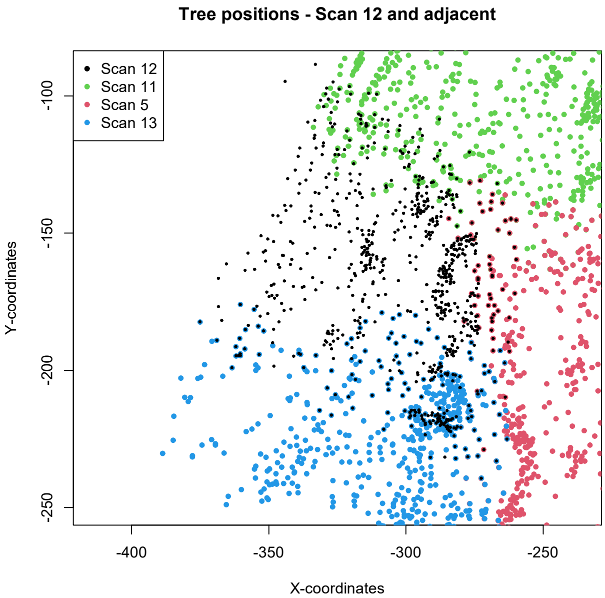


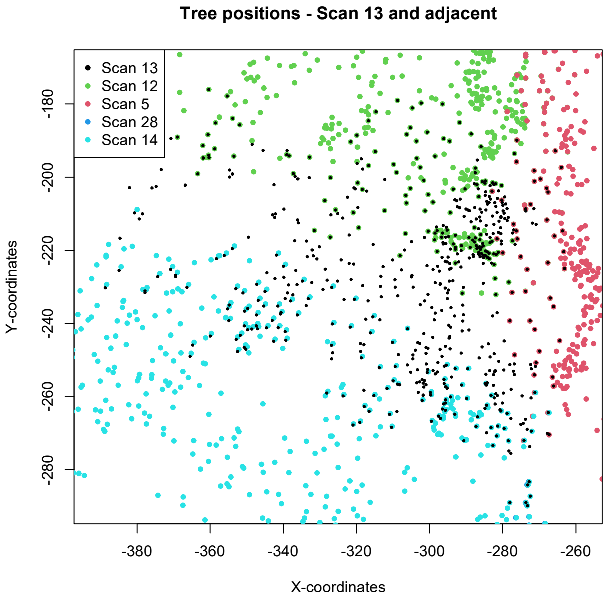
Tree positions - Scan 6 and adjacent

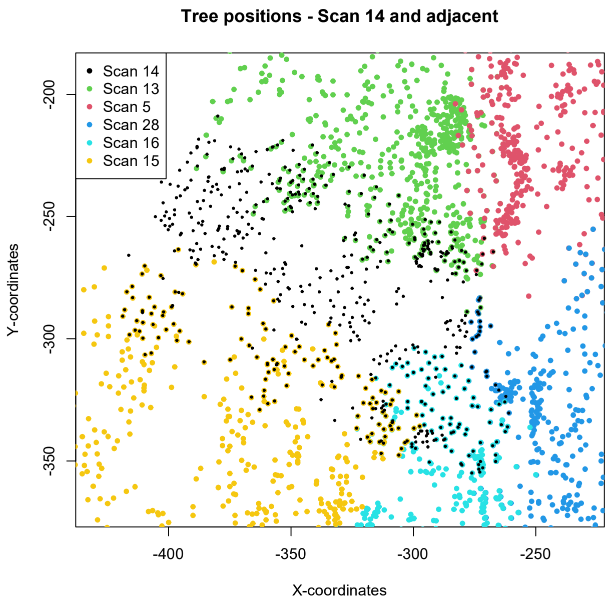


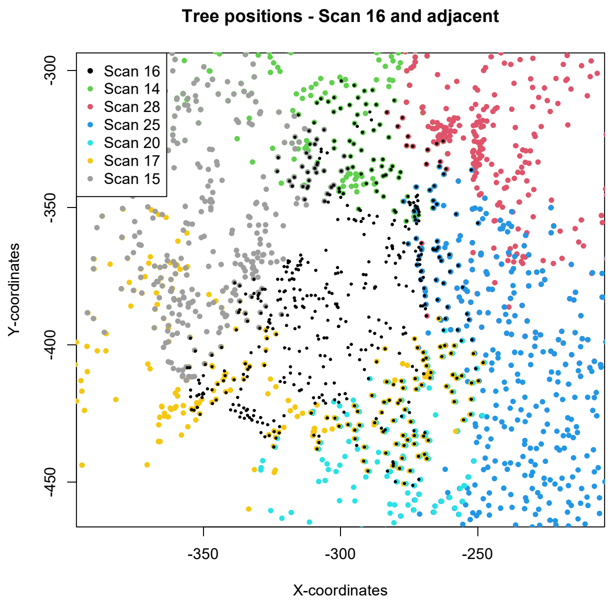

Tree positions - Scan 10 and adjacent
Scan 10
Scan 8
Scan 9
Scan 11
Y-coordinates
50
0
-50
-100
-150
-300
-250
-200
-150
-100
-50
X-coordinates

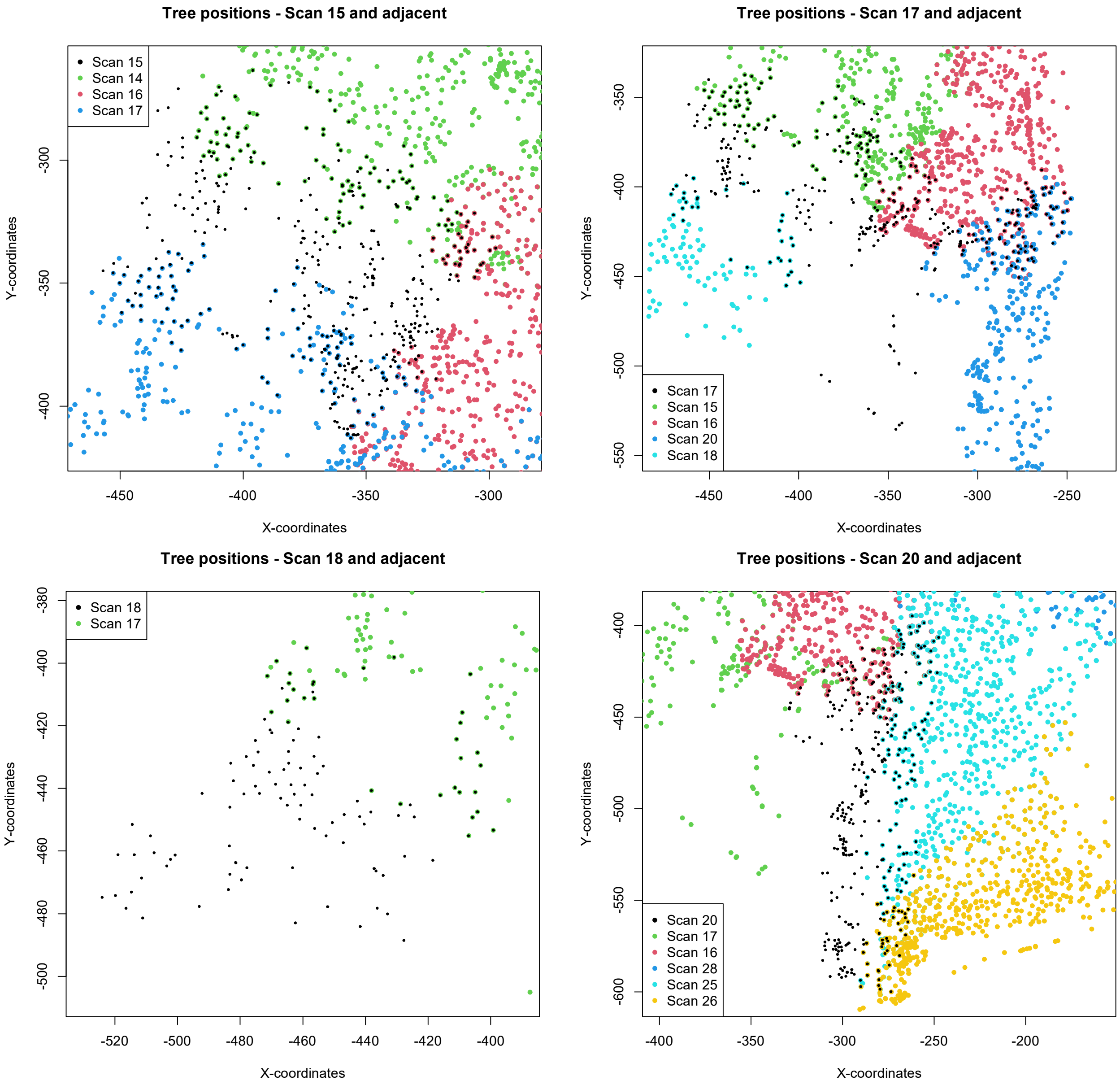

Tree positions - Scan 11 and adjacent
Scan 11
Scan 10
Scan 9
Scan 5
Scan 12
Y-coordinates
0
-50
-100
-150
-200
-350
-300
-250
-200
-150
-100
X-coordinates


Tree positions - Scan 12 and adjacent
Scan 12
Scan 11
Scan 5
Scan 13
Y-coordinates
-100
-150
-200
-250
-400
-350
-300
-250
X-coordinates


Tree positions - Scan 13 and adjacent
Scan 13
Scan 12
Scan 5
Scan 28
Scan 14
Y-coordinates
-180
-200
-220
-240
-260
-280
-380
-360
-340
-320
-300
-280
-260
X-coordinates


Tree positions - Scan 14 and adjacent
Scan 14
Scan 13
Scan 5
Scan 28
Scan 16
Scan 15
Y-coordinates
-200
-250
-300
-350
-400
-350
-300
-250
X-coordinates


Tree positions - Scan 16 and adjacent
Scan 16
Scan 14
Scan 28
Scan 25
Scan 20
Scan 17
Scan 15
Y-coordinates
-300
-350
-400
-450
-350
-300
-250
X-coordinates



Tree positions - Scan 15 and adjacent
Scan 15
Scan 14
Scan 16
Scan 17
Y-coordinates
X-coordinates
Tree positions - Scan 17 and adjacent
Scan 17
Scan 15
Scan 16
Scan 20
Scan 18
Y-coordinates
X-coordinates
Tree positions - Scan 18 and adjacent
Scan 18
Scan 17
Y-coordinates
X-coordinates
Tree positions - Scan 20 and adjacent
Scan 20
Scan 17
Scan 16
Scan 28
Scan 25
Scan 26
Y-coordinates
X-coordinates