Landslides (2021)18:3547–3558
DOI 10.1007/s10346-021-01723-4
Received: 9 April 2021
Accepted: 17 June 2021
Published online: 20September 2021

Luke Weidner · Megan van Veen · Matt Lato · Gabriel Walton

An algorithm for measuring landslide deformation in terrestrial lidar point clouds using trees

Abstract Terrestrial lidar data is a powerful resource for monitoring geohazards such as rockfall and landslides. However, vegetated landslides with horizontal shear surfaces remain difficult to characterize accurately due to a lack of exposed and appropriately oriented surfaces with respect to the lidar scanner. As an alternative, the movements of objects on the slope, such as trees, can be used to estimate slope movement. This paper demonstrates a novel semi-automated algorithm to extract and calculate the 3D displacement of trees on a slow-moving landslide, enabling more detailed landslide deformation analysis using point clouds. The method first uses local geometric descriptors to identify raw points corresponding to tree trunks. Several machine learning techniques are compared for this step, and an unsupervised decision tree algorithm with an accuracy of 84% is selected as the final method. After clustering points into individual trunks, trunks are matched through time according to several matching criteria, and their movement and rotation are then calculated using an iterative closest point algorithm. Accuracy of the automated displacement calculations is confirmed through comparison with manual point cloud measurements. A case study is presented demonstrating the method on a landslide through different seasonal vegetation changes, illustrating that algorithm parameters can be quickly adjusted to account for variations in tree species, density of foliage, scanner distance, and other factors. The final matching precision is demonstrated to be between 89 and 99%, indicating a very small number of false-positive matches.

Keywords Lidar · Tree matching · Vegetation

Introduction

Terrestrial lidar is increasingly being used to monitor slow-moving landslides in many types of terrain (Jaboyedoff et al. 2018; Booth et al. 2020). Increased use of this technology among geological professionals has the potential to allow for more comprehensive, data-driven natural hazard risk assessments, potentially saving time, resources, and lives (Lato et al. 2019). In recent applications, terrestrial lidar change detection is performed directly using the lidar point clouds, without the need for meshing, resulting in higher accuracy measurements in complex topography (Lague et al. 2013; Weidner and Walton 2020). Using terrestrial lidar data to track slow-moving landslides can be challenging, for several reasons. Such challenges include systematic errors introduced by registration of multi-temporal datasets and a lack of regular surfaces to measure due to dense vegetation cover (Kromer et al. 2017; Williams et al. 2018, 2021; Berg et al. 2020).

van Veen et al. (2019) noted that many slow-moving landslides in moderate-to-humid climates are partially or entirely covered in vegetation and often the failure surface is close to horizontal. In such cases, developing an accurate understanding of landslide velocities is very difficult with existing lidar processing techniques. While vegetation can be filtered from multi-return aerial lidar datasets to generate bare-earth topography, there are some geometrical limitations to measuring displacement on sub-horizontal translational landslides using bare-earth aerial lidar data (Lato et al. 2019). The limit of detectable change achieved using aerial lidar change detection is typically worse than terrestrial methods. van Veen et al. (2019) demonstrated that in such situations, movement of tree trunks can be used to track horizontal or sub-horizontal landslide deformation on vegetated slopes using terrestrial lidar scanning. While it was noted that tree movement could be caused by other effects, such as wind, snow, and tree growth, and that tree tilting due to landslide movement could be partially random, preliminary results indicated the method produces reasonable estimates of landslide deformation.

A limitation of the tree-tracking methodology as implemented by van Veen et al. (2019) is that it requires the manual extraction and processing of individual trees in the multi-temporal point clouds, which is a time-consuming iterative process. While point cloud processing is becoming increasingly automated (Bonneau et al. 2018, 2019; Schovanec et al. 2021), there is a lack of algorithms being developed and validated for geoscience and geological engineering applications (Weidner et al. 2019; Bonneau et al. 2020; Farmakis et al. 2020). In particular, extraction of meaningful objects in the scene, such as tree trunks, is still mainly a manual task, despite decades of research on machine learning–based semantic segmentation algorithms (Xie et al. 2020).

The current study presents a novel algorithm for the extraction and analysis of tree trunks in point clouds ("TreeTracker"), including the semantic segmentation (i.e., classification) of individual tree trunks, matching of corresponding trees in multi-temporal datasets, and calculation of the deformation of trees over time. Machine learning–based and rule-based (unsupervised) classification algorithms for tree trunk identification are compared, illustrating the usefulness of well-established shallow machine learning approaches as opposed to complex methods with high data requirements, such as deep learning.

The TreeTracker algorithm is developed and validated using a set of six terrestrial lidar datasets collected over a 2.5-year period for a partially vegetated landslide on the Peace River in western Canada (location 1, see Fig. 1). The landslide has a volume estimated at 1 million m^3 and is interpreted to be a single reactivated, very slow-moving, translational earth slide with a shear surface occurring along bedding planes in the underlying shale bedrock (Cruden and Varnes 1996; van Veen et al. 2019). Since the moving landslide mass is partially covered in trees, it was determined to be a good candidate for using the TreeTracker algorithm. As reported in van Veen et al. (2019), the slope for location 1 was scanned from the opposite side of the river valley six times between May 2018 and October 2020.

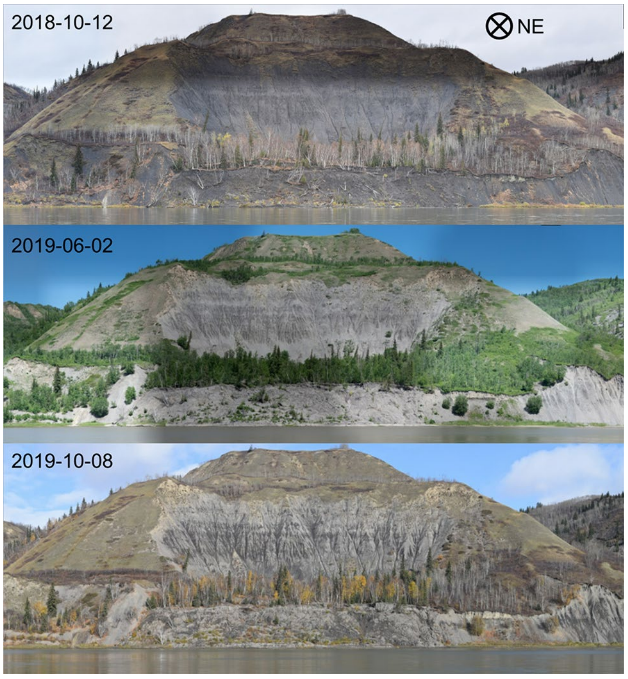


Fig. 1 Vegetated landslide used to develop and test the TreeTracker algorithm. Photo credit M. van Veen

TreeTracker workflow overview

The goal of the TreeTracker algorithm is to automatically identify and calculate the deformation between matching pairs of tree trunks from sequential lidar point clouds (individual point clouds are subsequently referred to as t1, time 1, and t2, time 2). Furthermore, the algorithm is intended to value the correctness of matches (minimize false positives) instead of extracting a complete inventory of trees on a slope (minimize false negatives). This is because not every single tree needs to be analyzed to capture the mechanisms of a landslide.

A visual workflow of the algorithm is presented in Figs. 2 and 3. The point clouds are first preprocessed in CloudCompare (CloudCompare 2021) to calculate geometric features, which are subsequently used to extract points corresponding to tree trunks. The only manual step required in the algorithm is for the user to visually confirm or adjust threshold values for four geometric features ("Extracting trunk points"). Then, a MATLAB script is run which loads both point clouds and performs the rest of the processing.

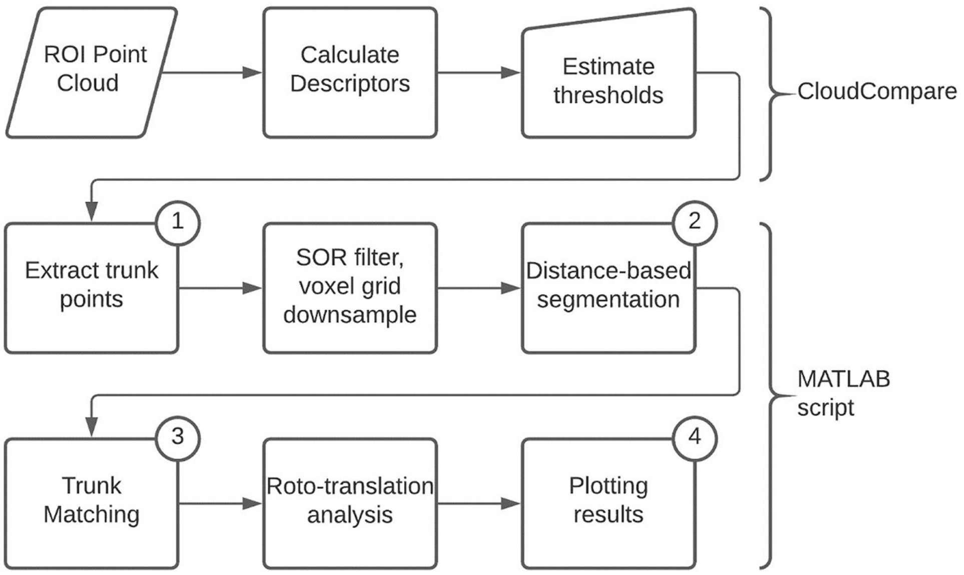
In the MATLAB script, a decision tree classification is first applied to both point clouds to extract tree trunks. A statistical outlier filter (number of neighbors = 20) and voxel grid down sample (grid step = 0.1 m) step are applied to the extracted trunks (step 1). We note that while these filter parameters are somewhat dependent on the original point density of the dataset, the total number of trunk matches is relatively unaffected by the specific values used. In a sensitivity test of the statistical outlier filter, varying the number of neighbors between 10 and 50 changed the number of matches by less than 5%.

The point clouds are then clustered into individual objects such that individual trunks or portions of trunks can be identified and referenced (step 2, "Trunk clustering"). In Fig. 3, pane 2, different groups of colored points represent different clusters. The matching portion of the workflow consists of several steps to take the tree trunks from t1, locate nearby clusters in the t2 point cloud, determine if the clusters are a match, then calculate and report the corresponding deformation (steps 3 and 4, "Trunk matching and roto-translation analysis"). Figure 3, pane 3, shows an example of a successful tree match between t1 (purple) and t2 (green), and pane 4 illustrates one method of presenting the results of the deformation calculation, where the red vectors represent the magnitude and direction of the tree movement between t1 and t2. The magnitude of deformation in units of meters is also displayed at the tail of each vector.

Extracting trunk points

The first step of the algorithm consists of semi-automatically separating points corresponding to tree trunks from other scene objects such as tree foliage and ground points. This can be

Fig. 2 Workflow diagram and illustration of key steps in the tree tracking algorithm. ROI, region of interest; SOR, statistical outlier removal filter



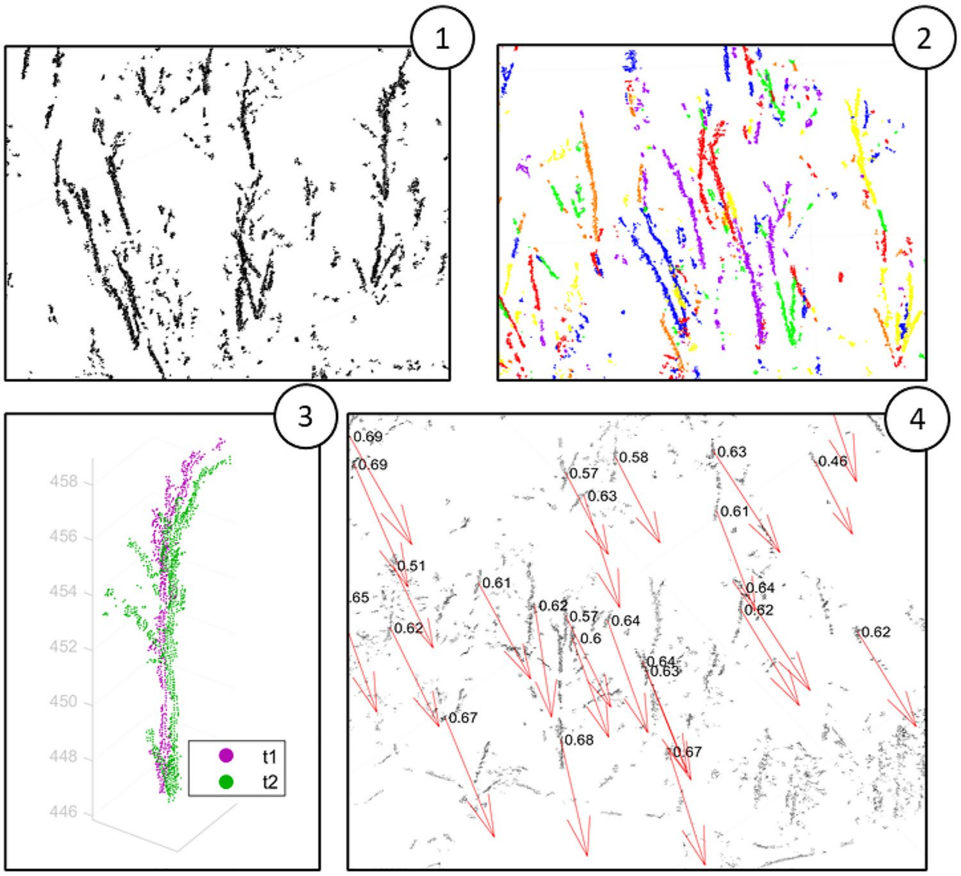


Fig. 3 Example output from each process step in the algorithm, corresponding to numbered steps in Fig. 2

accomplished using a number of methods, ranging from manual segmentation, calculating one of several discriminative geometric features, or through a variety of machine learning or deep learning approaches. For the segmentation of trees or tree parts, several specialized algorithms have previously been developed (Guan et al. 2015; Bonneau et al. 2020; Wang et al. 2020; Wu et al. 2020). In this work, we compare three classification techniques, including both supervised (requiring "training data") and unsupervised approaches, and choose one for the final algorithm that has a balance of accuracy and simplicity. The reason for comparing multiple approaches is that some algorithms may have very high accuracy but are unable to generalize to dissimilar site conditions without substantial modification or sizeable training datasets (Weidner et al. 2019, 2020). Therefore, we included methods that can be quickly modified based on visual inspection and adjusted to site-specific variations in tree and point cloud properties where needed.

To assess the accuracy of multiple classification approaches, a validation dataset was manually created with two classes: trunk and not trunk. The "not trunk" class included both tree foliage points and ground points. Two scans from a large, partially vegetated landslide were used (see "Case study: Peace River, Canada"): one from October 12, 2018, and one from June 2, 2019 (Fig. 4). These two datasets were selected for their differences in foliage between summer and fall, so that the classification approaches would need to be able to successfully classify both variations in data. Results for supervised algorithms are provided based on training on the June

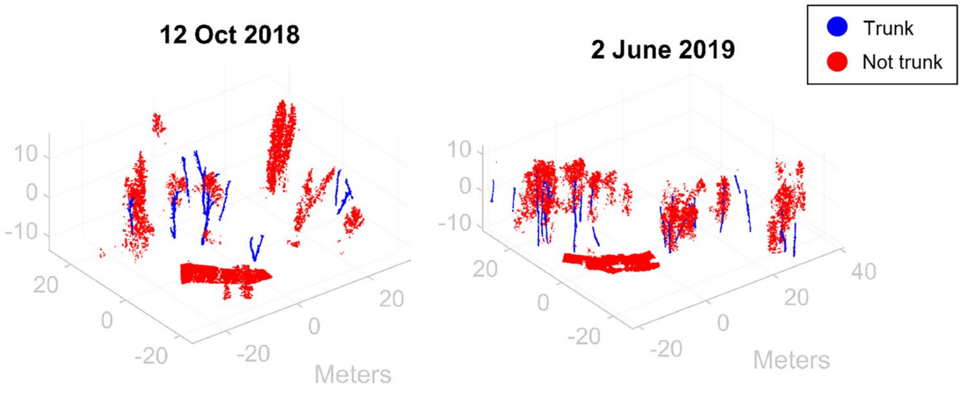


Fig. 4 Subsets of the labeled point clouds from two different times. Blue points are the trunk class and red points are the "not trunk" class, representing foliage and ground points

Table 1 Description of all classification methods, their geometric features, and additional algorithm parameters (if any)

Model	Features used	Parameters
Threshold	Linearity ($r = 0.5$)	1 feature threshold (visually estimated)
Decision tree	Linearity set	Max decision splits: 4
Decision tree	All	Max decision splits: 4
Random forest	Linearity set	60 trees, max decision splits per tree: 20
Random forest	All	60 trees, max decision splits per tree: 20
Designer tree	Designer set	4 feature thresholds (visually estimated)

2019 dataset and testing on the October 2018 dataset; however, it should be noted that reversing the training and testing order produced similar results. Throughout the training and testing process, the labeled point clouds were subsampled such that both classes (trunk/not trunk) had the same number of points, reducing the potential for bias in the classification results.

Point features and classification methods

We tested three types of point cloud classification with varying complexity: threshold, decision tree, and random forest (RF) (Tables 1 and 2). All three methods rely on "geometric" features to distinguish between different objects. Geometric features are based on the eigenvalues of the covariance matrix calculated for a small neighborhood of points, which can be manipulated to represent indicators of linearity, planarity, sphericity, and other descriptors. Linearity, for example, is a measure between zero and one of the degree to which the point arrangement in the neighborhood is 1D, while planarity similarly measures two-dimensionality, etc. This procedure is equivalent to performing a principal component analysis (PCA) on the local point neighborhood and calculating different ratios of the principal components. The reader is referred to the cited previous work for mathematical derivations of the geometric features (Brodu and Lague 2012; Weinmann et al. 2017).

Table 2 Description of geometric feature sets used in trunk classification techniques. The total number of features in the "All" feature set is 55

All (5 radii)	Linearity set	Designer set
Volume density	Linearity ($r = 1.5$)	Linearity ($r = 1.0$)
Omnivariance	Linearity ($r = 1.0$)	Linearity ($r = 0.5$)
Eigenentropy	Linearity ($r = 0.75$)	Verticality ($r = 0.5$)
Anisotropy	Linearity ($r = 0.5$)	Volume density ($r = 0.25$)
Planarity	Linearity ($r = 0.25$)	
Linearity		
PCA1		
PCA2		
Surface variation		
Sphericity		
Verticality		

The first test applied a threshold method. In the threshold method, a single geometric feature was selected based on the author's experience (linearity calculated with a search radius of 0.5 m), as able to discriminate tree trunks from other objects. All geometric features were calculated using the CloudCompare command line software (CloudCompare 2021). After the linearity has been calculated, points are classified as trunk or not trunk based on a threshold value manually selected using the "Display range" slider in CloudCompare (threshold value = 0.75 in the case of this study).

The second test applied a decision tree method. A supervised decision tree classifier was created using the "fitctree" function in MATLAB with a maximum number of decision splits set to 4. This creates a shallow decision tree that is quickly interpretable by the user. Two variants of a decision tree are fit to the training data: one using a large multiscale feature set of 55 features (11 feature types and five scales, see Table 2) and one using five linearity features at multiple scales. This allows us to estimate whether other features besides linearity are important to trunk identification, or if a strong classifier could be constructed with only linearity. In this case, scales represent the point neighborhood search radii used to calculate geometric features for each point in the dataset. For this study, radii of 0.25, 0.50, 0.75, 1.00, and 1.50 m were chosen (similar to Bonneau et al. 2020).

In addition to the data-driven decision tree described above, we constructed a manual "designer" decision tree based on observation of the features and scales that were selected consistently in other methods. The structure of the decision tree is shown in Fig. 5. The set of four features is shown in the "Designer set" column of Table 2. This method was tested because, similar to the threshold method, the decision split values can be quickly selected by the user to adjust for site-specific factors and potential seasonal variations in foliage cover. A sensitivity analysis on the threshold values was also performed to evaluate which features most strongly influence the classification results, and to determine whether threshold values visually estimated by the user are reasonably accurate.

The last classification method that was tested is a random forest, which is a supervised algorithm consisting of multiple uncorrelated decision trees. Each decision tree is trained on a random subset of features and training samples, and the final prediction is a majority vote of all the individual decision trees. To train a random forest, we used the "TreeBagger" function in MATLAB with 60 trees and a maximum number of decision splits per tree of 20. This is the most complex and least interpretable method, but it is also one of the most common and highest accuracy algorithms used for classification problems, and is therefore a useful point of comparison for the simpler methods (Fernandez-Delgado et al. 2014; Huang et al. 2020).

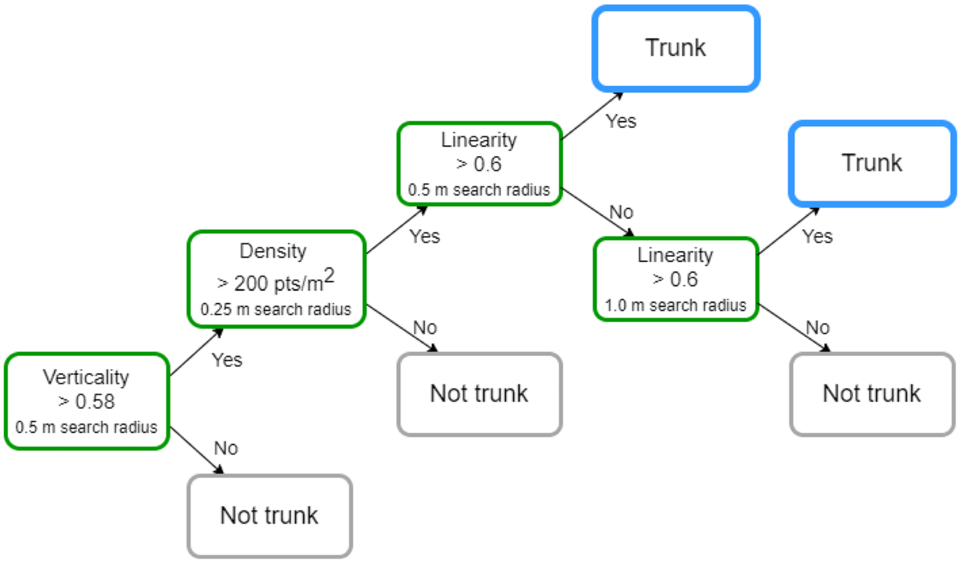


Fig. 5 Designer decision tree created using a subset of discriminative features

Trunk clustering

After extraction of trunk points, the point clouds were clustered such that individual trunk objects could be segmented, and their centroids calculated. For this step, the "pcsegdist" function in MATLAB was used, which is a distance-based clustering algorithm optimized for point clouds (see MATLAB documentation for additional details). The algorithm requires a distance parameter defining the minimum distance between any two points in different clusters (Fig. 6). If the distance parameter is too large, multiple trees could be joined in a single cluster, whereas if the parameter is too small, a single trunk could be segmented into several short clusters. We found that values for the distance parameter in the range of 0.4 to 0.7 m produced satisfactory results for both locations tested, but we note that for sites with different tree diameters and different distances between adjacent trees, this parameter may need to be adjusted. It is also noted that in the datasets used, many trees are only partially visible to the terrestrial lidar scanner and only a few partial segments of the trunk are captured in the point cloud. In these cases, the clustering algorithm is unable to cluster the entire tree for any reasonable value of the distance parameter. These partial clusters can still be used in tree deformation tracking, but the smaller the cluster, the more difficult it becomes to be confident if it represents an actual trunk.

After clustering, the centroid of each cluster is calculated, and clusters from the t1 cloud are sorted from largest to smallest based on the number of points. The sorting step was included based on the observation that larger clusters are more likely to represent actual tree objects, whereas small, sparse clusters likely represent "clutter," such as foliage that was mislabeled in the classification step or portions of branches. By performing subsequent steps of the analysis on the largest clusters only, we are able to improve the efficiency and accuracy of the algorithm.

Trunk matching and roto-translation analysis

After clusters have been calculated in both the t1 and t2 point clouds, clusters are searched for matches and the displacements between t1 and t2 are calculated (Fig. 7). The goal of the trunk matching algorithm is to minimize the number of false-positive matches while keeping a reasonable number of true positives. Therefore, it is acceptable if some proportion of correct matches are rejected, as long as the remaining matches are well-distributed across the area of interest. The algorithm developed for identifying matches operates as follows: First, starting with a candidate cluster from t1, the t2 cloud is searched for the nearest neighbor cluster centroid. Then,

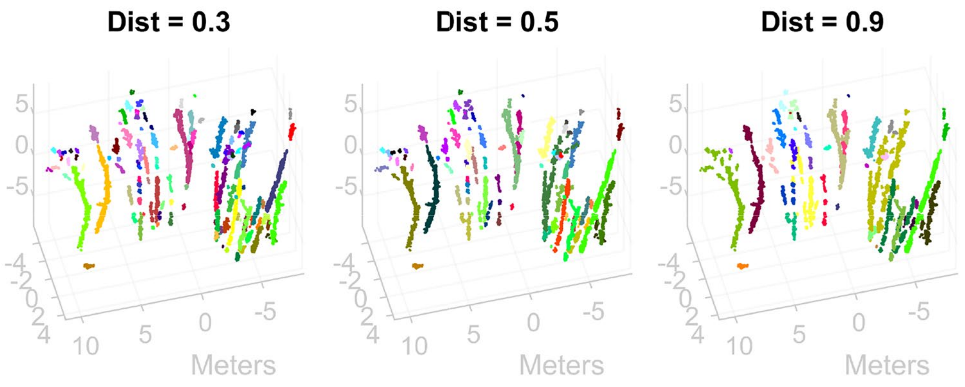


Fig. 6 Example of clustering results with three different values for the distance parameter. Groups of points with the same color represent individual clusters. A small value (left panel) results in some trees being separated into many small clusters, which is not useful for deformation tracking. A large value (right panel) groups multiple trunks and some clutter together in the same cluster

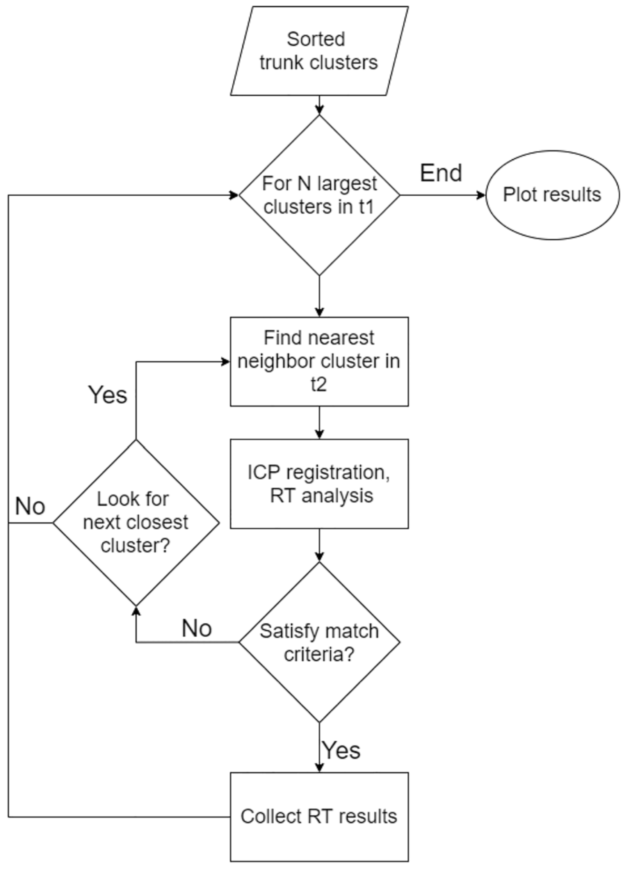

Sorted trunk clusters
For N largest clusters in t1
End
Plot results
Find nearest neighbor cluster in t2
Yes
ICP registration, RT analysis
No
Look for next closest cluster?
No
Satisfy match criteria?
Yes
Collect RT results

Fig. 7 General structure of the trunk matching algorithm

the displacement between the two clusters is calculated according to the methodology developed by Oppikofer et al. (2009). In this "roto-translation (RT)" method, the iterative closest point (ICP) registration algorithm is used to calculate a best-fit rigid-body transformation between two point clouds, and the resulting 3D transformation matrix defines the movement and rotation in space from one point cloud to the other. In our algorithm, we used an inlier ratio of 0.5 for the ICP alignment to reduce the effect of far-away points and noise on the registration. The 3D transformation matrix can be converted to *X*, *Y*, and *Z* components of displacement, the azimuth and plunge of displacement, the angle of toppling, and the angle and axis of rotation.

Once a pair of clusters has been registered and the deformation calculated, it is subjected to a series of criteria to determine whether it is a successful match (i.e., it is the same tree in two different point clouds). The four criteria used to filter potential matches, all of which must be met simultaneously for the match to be recorded, are summarized in Table 3. Examples of a match that would pass all criteria, one that would fail the "Zdiff" criteria, and one that would fail the root mean square error (RMSE) criteria, are presented in Fig. 8. The "Tdist" criterion sets a limit on the distance a single tree can "move" for it to be accepted as a match. Cluster pairs with a large total distance relative to the expected magnitude of landslide movement are likely incorrect, and accordingly, the value of this parameter can be set to a relatively conservative high value based on prior knowledge of the site. The "Zdiff" criterion ensures that the pair of clusters has similar total heights and occur at similar elevations, while the "MinHeight" criterion additionally sets a minimum total height that both clusters must have. The RMSE criterion exploits the ICP registration algorithm to measure how closely the pair of clusters "match" in terms of their overall shape. A higher RMSE reported by ICP indicates the two point clouds were unable to be registered closely, giving a powerful overall measure of the quality of the match. A preliminary threshold of 0.05 m was selected based on observing the RMS errors of many correct and incorrect matches. However, this value will vary depending on quality of the lidar data (resolution, distance from the slope, atmospheric effects, etc.).

If a cluster pair passes all four criteria, its RT matrix and corresponding deformation values are recorded. Otherwise, the pair is rejected, and the entire matching process is repeated for the next candidate cluster in t1, up to some set number of candidate clusters. Optionally, the user can also set that a certain number of the nearest neighbor clusters should be searched for each t1 candidate, such that if the first nearest cluster is not a match, the next nearest clusters, up to some set number, can also be checked. Our observation in a case study (see "Case study: Peace River, Canada") was that candidate cluster pairs were either a successful match with the first nearest neighbor or with none of the 5 nearest neighbor clusters. Accordingly, this option did not result in an increase in successful matches, but it could potentially be useful at other sites if the distance between adjacent trees is less than the magnitude of landslide displacement.

Validation of trunk matching

To provide a quantitative evaluation of trunk matching accuracy and the sensitivity of the four matching criteria, a manual validation dataset at location 1 (May 11, 2018, to October 12, 2018) was created. A set of 342 candidate matches were labeled by the author as either being a correct match or an incorrect match. This resulted in a dataset of 146 correct matches and 196 incorrect matches. Based on preliminary

Table 3 Matching criteria used to identify correct trunk matches

Criteria	Explanation	Example value
Tdist	Maximum total distance between clusters	1.5 m
Zdiff	Maximum difference in minimum and maximum height between clusters	1.5 m
Root mean square error (RMSE)	Maximum ICP RMSE	0.05 m
MinHeight	Minimum height of both clusters	1.5 m

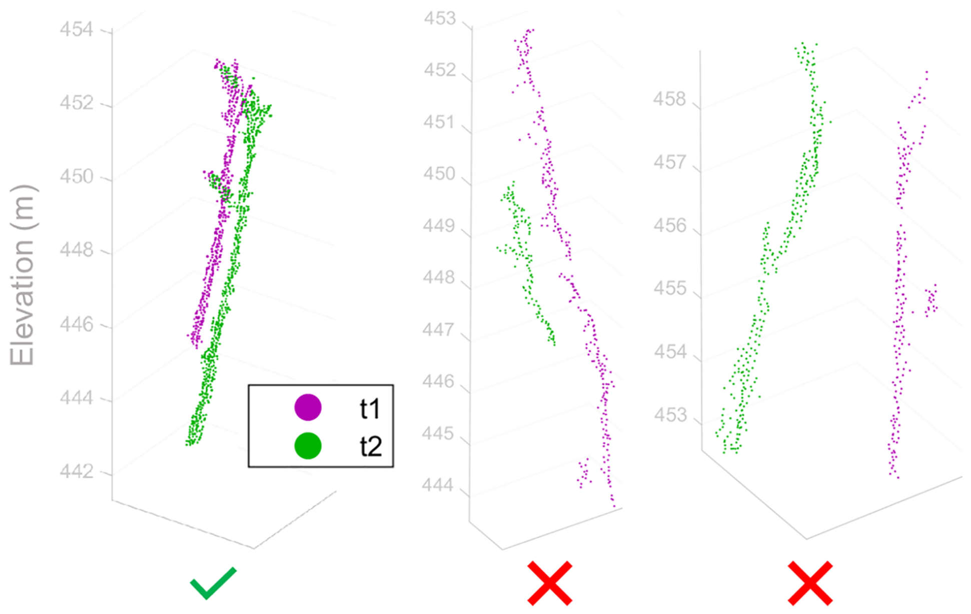


Fig. 8 Illustration of correctly matched (left) and two incorrectly matched (center and right) point cloud pairs. The correct match has an ICP RMSE of 0.033 m and passes the other criteria. The center match does not pass the Zdiff criterion and the right match does not pass the RMSE criterion. Units are in meters

testing, the Tdist and MinHeight criteria were determined to have a negligible effect on overall matching accuracy. For the two remaining criteria, Zdiff and RMSE, a grid search was performed where combinations of values were used, and the numbers of true-positive and false-positive matches were recorded for each combination. As mentioned above, true negatives and false negatives were deemed not relevant for this analysis, as the main benefit of the TreeTracker algorithm is to track trees with high precision and with a wide distribution across the slope, not necessarily to extract and track all trees in the point cloud.

In addition to evaluating matching accuracy, displacement values between the May 11, 2018, and October 12, 2018, datasets were checked against manual calculations performed by van Veen et al. (2019). In the manual workflow, the same basic steps of segmenting tree trunks and performing the ICP RT are performed, but in an entirely manual manner using PolyWorks and MATLAB software (InnovMetric 2019).

Results

Trunk point classification

Classification accuracies for all methods are summarized in Table 4. The overall accuracy metric is defined as the number of correct predictions (points classified correctly as trunk or not trunk) divided by the total number of points in the point cloud. It is observed that the most complex classification method, the random forest with all 55 features, had the highest accuracy at 86.2%, and the simplest method, the 0.5 m linearity threshold, had the lowest accuracy at 77.6%. Interestingly, the designer tree, which uses only four features can be adjusted manually for site-specific parameters, achieved an accuracy of 84.2%, close to that of the most complex method. While higher accuracies could likely be achieved with more complex methods, the accuracy of the designer tree was deemed acceptable for trunk matching purposes because, in addition to the fact that it is interpretable and flexible, an approximately 15% error rate was found to not impact final trunk matching performance. In other words, filtering steps applied elsewhere in the workflow sufficiently correct for errors in the initial classification.

Figure 9 presents the results of the sensitivity analysis for the designer tree. The results show that accuracy is moderately sensitive to the threshold values of the designer tree, with linearity features having the greatest impact on the overall results. However, visually estimated threshold values were ultimately at or near their optimal values, indicating that the user should be able to quickly select appropriate values simply by visual estimation. For example, the verticality (0.5 m search radius) threshold was visually estimated to be 0.58, and in the sensitivity analysis, this threshold value was found to be the optimal value.

It is also important to note that seasonal changes were observed to have a significant effect on the optimal threshold values, particularly for linearity. In June 2019, a linearity (0.5 m search radius) threshold of 0.6 was near-optimal, but for October 2018, this value was too low. This is because the increased foliage in the June 2019 dataset typically decreased the small-scale linearity of trunks surrounded by foliage, and so a lower linearity threshold was needed to identify trunk points in that dataset.

Table 4 Overall accuracies for trunk vs. not trunk classification for all model types tested

Model	Overall accuracy
Linearity threshold	77.6%
Decision tree (linearity set)	77.4%
RF (linearity set)	82.4%
Decision tree (all)	81.7%
RF (all)	86.2%
Designer tree	84.2%

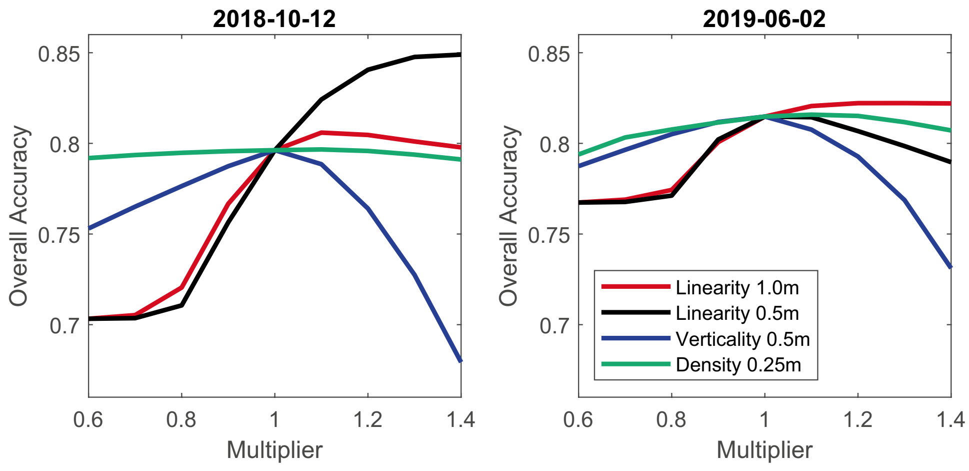


Fig. 9 Sensitivity analysis of geometric feature thresholds for the "designer" decision tree. Note that the same set of initial thresholds (i.e., values at a multiplier of 1) were used for both the 2018 and 2019 datasets, and therefore, the values are not necessarily optimal for both sites

Trunk matching

Trunk matching performance evaluations for various combinations of the RMSE and Zdiff criteria are shown in Table 5. Precision, defined as the number of true positives divided by the sum of true positives and false positives, was chosen as the performance metric. This was because the overall goal of the algorithm is not necessarily to extract *all* trees in the dataset, but rather to extract enough to characterize slope movement. Therefore, we ignore false negatives and only consider true positives and false positives. It is evident that depending on the parameters, a large range of successful matches and false positives can be found, with corresponding precision values ranging from 89 to 99%. False positives are most sensitive to the RMSE criteria; as the maximum RMSE value is lowered, the number of false positives sharply declines with a comparatively small reduction in the number of true positives.

We can also visualize the distribution of incorrect tree matches in the data. Figure 10 presents deformation magnitudes and direction vectors for the 2018 validation dataset and two possible combinations of the RMSE and Zdiff criteria (cells with asterisks in Table 5). Each point represents a matched tree, and incorrect matches are highlighted with triangles. The 91% precision case shows many matches were captured by the algorithm (91), but a correspondingly higher number of false positives are included as well (9). The 96% precision case represents the most stringent set of thresholds tested, resulting in27 true positives and 1 false positive. In this case, the spatial distribution of matches is less complete, and some of the trunks with the highest deformation are excluded. However, only a single false-positive match was observed using these criteria. The highest overall precision of 99% was achieved by increasing the Zdiff criteria to 2.5 m, allowing for greater differences in height between clusters. This resulted in an increase in true positives with no increase in false positives.

False positives can sometimes be identified visually based on their having very different directions and magnitudes from their neighbors. A potential addition to the TreeTracker algorithm, which was not explored in this work, would be to flag matches as suspected false positives based on comparison with their spatial neighbors. However, many other false positives appear to have similar directions and magnitudes to their neighbor true-positive matches,

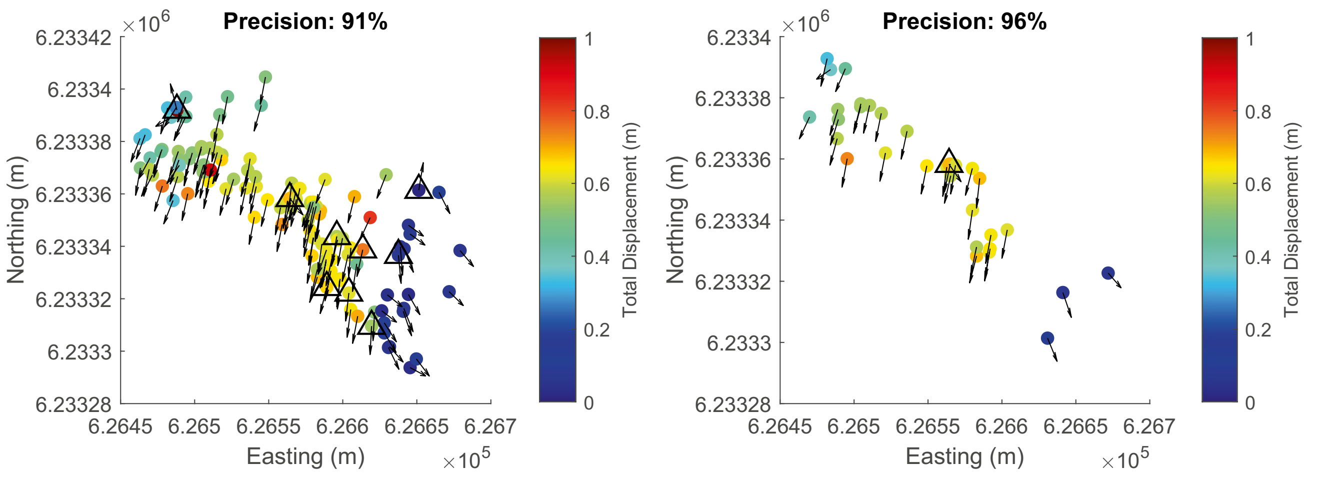


Fig. 10 Map of deformation magnitudes and directions for two scenarios of matching criteria (asterisks in Table 5). Vectors indicate movement direction, while colors indicate displacement magnitude. Black triangles represent false-positive matches

Table 5. Trunk matching performance results for several combinations of the Zdiff (columns) and RMSE (rows) criteria. Cells with asterisks are plotted in Fig. 10.

RMSE	False Positives						
0.05	11	11	9	8	8	4	3
0.045	5	5	4	3	3	1	1
0.04	2	2	1	1	1	1	1
	True Positives						
0.05	113	101	91	80	67	51	33
0.045	102	92	83	74	63	47	31
0.04	81	74	68	59	52	39	27
	Precision						
0.05	91%	90%	91%*	91%	89%	93%	92%
0.045	95%	95%	95%	96%	95%	98%	97%
0.04	98%	97%	99%	98%	98%	98%	96%*
	3.5	3	2.5	2	1.5	1	0.5
	Zdiff value						

and these cases would not be able to be automatically identified. Furthermore, depending on the mechanism of landslide displacement, a difference in direction and magnitude may not be unexpected if some trees are rotating and tilting backwards, as others are translating down the slope.

In observing the actual movement magnitudes and directions, it is noted that results agree with those reported by van Veen et al. (2019) for the same dataset but using manual point cloud processing tools. van Veen et al. (2019) reported that deformations generally ranged from 0.5 to 0.8 m and the mean trend of movement was approximately 200° azimuth. Using the TreeTracker algorithm, a range of deformations from 0 to 0.74 m were calculated, with a mean trend of 195°.

Case study: Peace River, Canada

The TreeTracker workflow was applied to the full set of five epochs (six scan dates) at location 1. While most data were collected in the spring and fall months to limit the amount of foliage obstructing the view of the slope, one dataset was collected in June 2019, resulting in a lower quality point cloud on that date. The algorithm parameters selected for the five epochs analyzed here are shown in Table 6. Initially, a single parameter set was used for all five pairs, but it was observed that pairs that included the June 2019 dataset had an undesirably low number of matches. Consequently, the classification and matching parameters were adjusted to account for the increased foliage in the June scan. The linearity thresholds were lowered, and the point density threshold was raised, while the RMSE and Zdiff matching criteria were relaxed. Despite these adjustments, the numbers of matches for these pairs of data were still lower than for pairs that did not include the June 2019 dataset.

Analysis of deformation results showed a surge in landslide movement between Spring and Fall 2018 (Fig. 11a), less movement during the winter months of 2018 to 2019 (Fig. 11b), and negligible movement on the slope overall between Spring 2019 and Spring 2020 (Fig. 11c, d). Between Spring and Fall 2020, a larger magnitude surge in movement between 1.0 and 1.5 m was observed (Fig. 11e). In this time period, some isolated deformations larger than 1.5 m were observed manually but were unable to be detected automatically by the algorithm due to large rotations. By applying automated methods to track trees across the entire slope, spatial patterns and the lateral limits of landslide movement could be identified. In particular, the southeast boundary of the moving mass is clearly visible during surge periods, and a slight gradation of increasing deformation from west to east is also evident.

Table 6 All TreeTracker parameters for comparisons at location 1. Thresholds are given in the same order as in Table 2 for the designer set

t1	t2	t1 thresholds	t2 thresholds	Tdist (m)	RMSE (m)	Zdiff (m)	MinHeight (m)	Number of matches
11-May-18	12-Oct-18	[0.55, 0.60, 0.58, 170]	[0.55, 0.60, 0.58, 170]	1	0.045	1	1.5	58
12-Oct-18	2-Jun-19	[0.55, 0.60, 0.58, 170]	[0.5, 0.50, 0.58, 180]	1	0.05	3	1.5	21
2-Jun-19	9-Oct-19	[0.5, 0.50, 0.58, 180]	[0.55, 0.60, 0.58, 170]	1	0.05	3	1.5	25
9-Oct-19	7-May-20	[0.55, 0.60, 0.58, 170]	[0.55, 0.60, 0.58, 170]	1	0.045	1	1.5	41
7-May-20	2-Oct-20	[0.55, 0.60, 0.58, 170]	[0.45, 0.60, 0.55, 200]	2	0.05	1	1.5	57

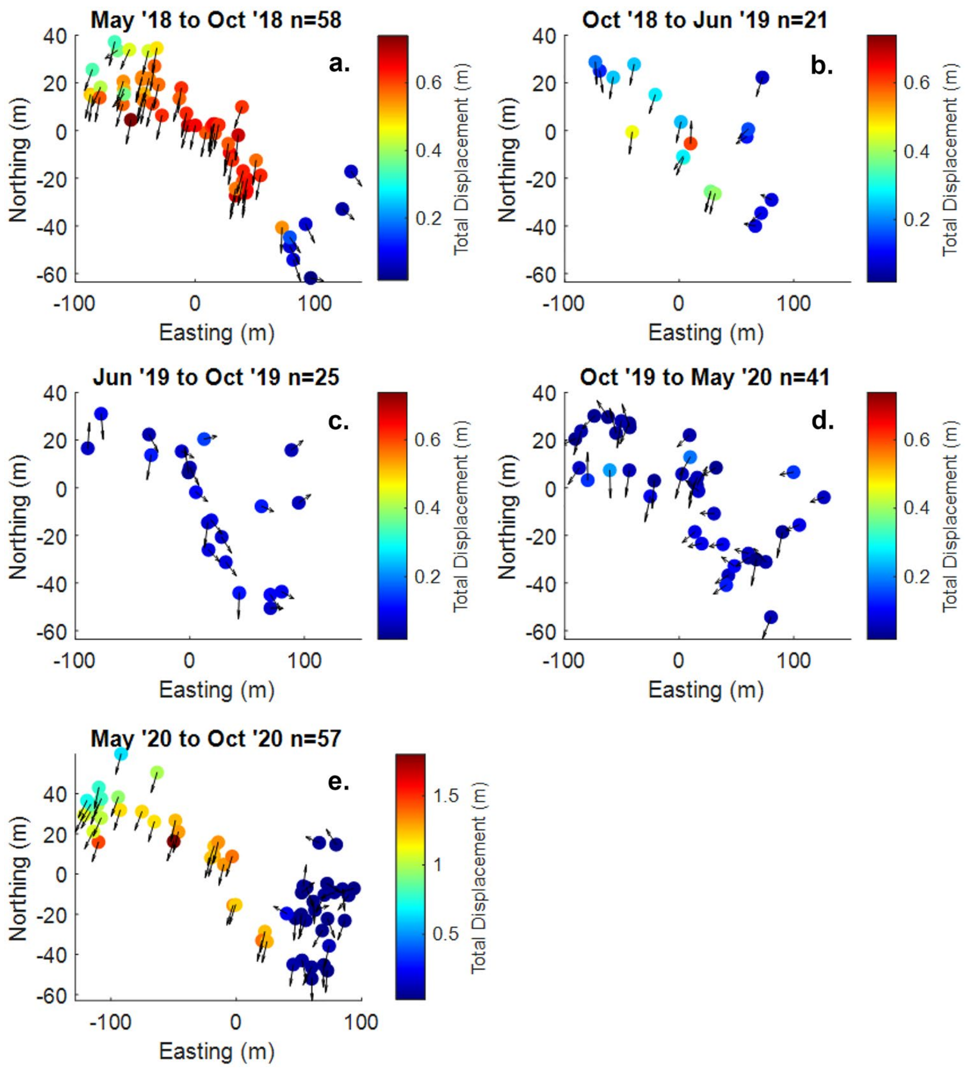


Fig. 11 Maps of tree displacement and direction of movement for five time steps between May 2018 and October 2020 at location 1. Each of the five panels represents a different epoch, with the epoch time span indicated in the title above each panel. Each colored point in a panel represents a matched tree in the epoch, and the number of matches (n) for each time step is also included in the title. Note that a different color scale has been used for the May 2020 to October 2020 period

Discussion

Classification methods

This paper compared several classification methods to identify tree trunk points in unstructured point clouds. While the supervised random forest model had the highest accuracy, it was also by far the most complex model with 55 input features and an ensemble of 60 decision trees. The use of a supervised classification algorithm such as random forest has potential for geological engineering applications, but ensuring that a trained algorithm is able to perform well in a variety of different geologic settings, climates, or data collection scenarios is difficult (Weidner et al. 2020). The composition of the training data has a large influence on generalization accuracy when the dataset size is small, even if the algorithm is not likely to be overfitting. We demonstrate that an unsupervised algorithm with four key features is easier for a non-expert to adjust in response to site-specific variations, and this only results in a small reduction in overall accuracy.

This also makes the classification step consistent with the other steps of the TreeTracker algorithm, in that parameters can be adjusted quickly in response to seasonal differences, different scanner types and ranging accuracies, different tree species, etc. The entire TreeTracker algorithm takes one to several minutes to run on a higher end laptop computer or typical desktop computer (4-core CPU, 16-GB RAM, dedicated GPU), meaning that several parameter combinations can quickly be tested to find a suitable balance between strict matching criteria and a reasonable number of matches.

Limit of detection

The deformation values calculated using the RT method and the ICP algorithm are subject to uncertainties, and therefore, a limit of detection needs to be estimated and used to filter potential errors from the final results. Tree growth, wind, and snow loading could cause real tree deformation that is not representative of landslide movement. However, in this study, we can deem snow or wind events as unlikely based on the translational character of the movements and corroboration with field observations. Dataset characteristics such as registration error and the position, distance, and quality attributes of the scanner also need to be considered. For example, in the case study presented in this paper, a registration error of 0.10 m was estimated using regions of the slope presumed to be outside the landslide extents and therefore stable. Registration errors need to be carefully documented with any application of the TreeTracker algorithm, as they could be misinterpreted as an indication of widespread slope movement and therefore bias displacement results. Additional errors are more difficult to estimate quantitatively, but it is expected that tree growth and ranging error would likely add several centimeters of uncertainty, resulting in an estimated limit of detection of around 0.15 m. This means that deformation magnitudes near to or less than this limit should be considered noise and not real movement of the slope.

In addition to this lower limit, a less well-defined upper limit also exists, wherein movements and/or rotations larger than a certain amount are likely to be missed by the algorithm. When landslide deformation is large relative to the spacing between adjacent trees, accurate trunk matching becomes more difficult. Furthermore, if a tree tilts by more than around 45°, it will be missed because it no longer satisfies the verticality constraint, and it may also fail the RMSE criterion due to trunk bowing or branches snapping. Overall, more quantitative evaluation of accuracy is needed to understand the potential sources of error in the TreeTracker algorithm. It is likely that the error may vary when TreeTracker is applied to other slopes.

Conclusions

Slow-moving landslides with sub-horizontal shear surfaces can be difficult to accurately characterize using terrestrial and aerial lidar scanning. Vegetation on landslides is typically seen as a nuisance from a slope monitoring perspective, but tree trunks captured in point cloud data can be used to estimate landslide deformation even if very few ground points are available. This paper demonstrates an algorithm to automatically calculate the movement of trees on landslides using a novel semantic segmentation and matching algorithm combined with the methodologies of van Veen et al. (2019) and Oppikofer et al. (2009) for roto-translation analysis. On the test dataset, tree trunks are matched with a precision (user's accuracy) ranging from 89 to 99% depending on parameters. Tree trunk deformation for tens of trees across a slope can be automatically estimated. The included case study illustrates that the developed system enables a new level of detailed landslide analysis using terrestrial lidar datasets. The code for this paper can be found at https://github.com/lmweidner/treetracker.

The core functionality of the TreeTracker framework could be extended to more types of applications in the future. In the context of vegetation tracking, TreeTracker could be applied to point clouds collected via airborne laser scanning (ALS), enabling analysis of tree displacements over an entire corridor. Other failure mechanisms could also be monitored, such as hillslope creep or settlement over longer time scales. A similar framework could be applied to other geohazards, such as rock slope monitoring (e.g., Oppikofer et al. 2009), or monitoring of structural elements such as bolts in excavations, but this would require modification of the classification module to identify objects other than trees.

References

Berg N, Hori T, Take WA (2020) Calculation of 3D displacement and time to failure of an earth dam using DIC analysis of hillshade images derived from high temporal resolution point cloud data. Landslides 17:499–515. https://doi.org/10.1007/s10346-019-01284-7

Bonneau D, DiFrancesco P-M, Hutchinson DJ (2019) Surface reconstruction for three-dimensional rockfall volumetric analysis. ISPRS Int J Geo-Inf 8:548. https://doi.org/10.3390/ijgi8120548

Bonneau DA, DiFrancesco P-M, Hutchinson DJ (2020) A method for vegetation extraction in mountainous terrain for rockfall simulation. Remote Sens Environ 251:112098. https://doi.org/10.1016/j.rse.2020.112098

Bonneau DA, Hutchinson DJ, Difrancesco PM, Coombs M, Sala Z (2018) 3-Dimensional rockfall shape back-analysis: methods and implications. Nat Hazards Earth Syst Sci Discuss 1–35. https://doi.org/10.5194/nhess-2018-366

Booth AM, McCarley JC, Nelson J (2020) Multi-year, three-dimensional landslide surface deformation from repeat lidar and response to precipitation: Mill Gulch earthflow, California. Landslides 17:1283–1296. https://doi.org/10.1007/s10346-020-01364-z

Brodu N, Lague D (2012) 3D terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: applications in geomorphology. ISPRS J Photogramm Remote Sens 68:121–134. https://doi.org/10.1016/j.isprsjprs.2012.01.006

CloudCompare (2021) cloudcompare.org, Version 2.12, Open Source Project

Cruden DM, Varnes DJ (1996) Landslides: investigation and mitigation. Chapter 3 - Landslide types and processes. Transp Res Board Spec Rep

Farmakis I, Bonneau D, Hutchinson DJ, Vlachopoulos N (2020) Supervoxel-based multi-scale point cloud segmentation using FNEA for object-oriented rock slope classification using TLS, in: ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Presented at the XXIV ISPRS Congress, Commission II (Volume XLIII-B2-2020) - 2020 edition, Copernicus GmbH, pp 1049–1056. https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-1049-2020

Fernandez-Delgado M, Cernadas E, Barro S, Amorim D (2014) Do we need hundreds of classifiers to solve real world classification problems? J Mach Learn Res 15:3133–3181

Guan H, Yu Y, Ji Z, Li J, Zhang Q (2015) Deep learning-based tree classification using mobile LiDAR data. Remote Sens Lett 6:864–873. https://doi.org/10.1080/2150704X.2015.1088668

Huang F, Cao Z, Guo J, Jiang S-H, Li S, Guo Z (2020) Comparisons of heuristic, general statistical and machine learning models for landslide susceptibility prediction and mapping. CATENA 191:104580. https://doi.org/10.1016/j.catena.2020.104580

InnovMetric (2019) Polyworks, Version 2250. InnovMetric

Jaboyedoff M, Abellán A, Carrea D, Derron MH, Matasci B, Michoud C (2018) 17. Mapping and monitoring of landslides using LIDAR, in: Natural hazards: earthquakes, volcanoes, and landslides. CRC Press pp 397–402. https://doi.org/10.1201/9781315166841

Kromer RA, Abellan A, Hutchinson DJ, Lato M, Chanut MA, Dubois L, Jaboyedoff M (2017) Automated terrestrial laser scanning with near real-time change detection-monitoring of the Séchilienne landslide. Earth Surf Dyn Discuss 1–33. https://doi.org/10.5194/esurf-2017-6

Lague D, Brodu N, Leroux J (2013) Accurate 3D comparison of complex topography with terrestrial laser scanner: application to the Rangitikei canyon (N-Z). ISPRS J Photogramm Remote Sens 82:10–26. https://doi.org/10.1016/j.isprsjprs.2013.04.009

Lato MJ, Anderson S, Porter MJ (2019) Reducing landslide risk using airborne Lidar scanning data. J Geotech Geoenvironmental Eng 145:06019004. https://doi.org/10.1061/(ASCE)GT.1943-5606.0002073

Oppikofer T, Jaboyedoff M, Blikra L, Derron M-H, Metzger R (2009) Characterization and monitoring of the Åknes rockslide using terrestrial laser scanning. Nat Hazards Earth Syst Sci 9:1003–1019. https://doi.org/10.5194/nhess-9-1003-2009

Schovanec H, Walton G, Kromer R, Malsam A (2021) Development of improved semi-automated processing algorithms for the creation of rockfall databases. Remote Sens 13:1479. https://doi.org/10.3390/rs13081479 van Veen M, Porter M, Lato M, Mitchell A, Whadcoat S (2019) Assessing landslide deformation using trees in terrestrial lidar data, in: Geo St. Johns 2019. Presented at the Canadian Geotechnical Conference. Can Geotech J

Wang D, Takoudjou SM, Casella E (2020) LeWoS: A universal leaf-wood classification method to facilitate the 3D modelling of large tropical trees using terrestrial LiDAR. Methods Ecol Evol 11:376–389. https://doi.org/10.1111/2041-210X.13342

Weidner L, Walton G, Kromer R (2019) Classification methods for point clouds in rock slope monitoring: a novel machine learning approach and comparative analysis. Eng Geol 263:105326. https://doi.org/10.1016/j.enggeo.2019.105326

Weidner L, Walton G (2020) Monitoring and modeling of the DeBeque Canyon landslide complex in three dimensions. Presented at the 54th U.S. Rock Mechanics/Geomechanics Symposium. Am Rock Mech Assoc

Weidner L, Walton G, Kromer R (2020) Generalization considerations and solutions for point cloud hillslope classifiers. Geomorphology 107039. https://doi.org/10.1016/j.geomorph.2020.107039

Weinmann M, Jutzi B, Mallet C, Weinmann M (2017) Geometric features and their relevance for 3D point cloud classification. ISPRS Ann. Photogramm. Remote Sens Spat Inf Sci IV-1/W1 157–164. https://doi.org/10.5194/isprs-annals-IV-1-W1-157-2017

Williams JG, Rosser NJ, Hardy RJ, Brain MJ, Afana AA (2018) Optimising 4-D surface change detection: an approach for capturing rockfall magnitude–frequency. Earth Surf Dyn 6:101–119. https://doi.org/10.5194/esurf-6-101-2018

Williams JG, Anders K, Winiwarter L, Zahs V, Höfle B (2021) Multi-directional change detection between point clouds. ISPRS J Photogramm Remote Sens 172:95–113. https://doi.org/10.1016/j.isprsjprs.2020.12.002

Wu B, Zheng G, Chen Y (2020) An improved convolution neural network-based model for classifying foliage and woody components from terrestrial laser scanning data. Remote Sens 12:1010. https://doi.org/10.3390/rs12061010

Xie Y, Tian J, Zhu XX (2020) A review of point cloud semantic segmentation. IEEE Geosci Remote Sens Mag 0–0. https://doi.org/10.1109/MGRS.2019.2937630

Luke Weidner (✉) · **Megan** van **Veen** · **Matt Lato** · **Gabriel Walton**

Department of Geology and Geological Engineering, Colorado School of Mines, CO, Golden, USA

Email: weidner@mines.edu