Tree Detection Around Forest Harvester Based on Onboard LiDAR Measurements

Satu Sihvo, Petra Virjonen, Paavo Nevalainen, Jukka Heikkonen
Faculty of Science and Engineering
Turku University
Turku, Finland
sahasi@utu.fi, petra.virjonen@utu.fi, paavo.nevalainen@utu.fi, jukka.heikkonen@utu.fi

Abstract—**This paper proposes a new approach for the detection of tree locations around forest machines producing a situational model based on on-site terrestrial LiDAR data collected during harvesting operation. A triangularized ground model is used to planarize the point cloud in order to simplify the tree detection. The planarized ground makes the vertical cutting of the point cloud systematical. Tree stem lines detected from individual trees at individual scan views are used to guide the final alignment into global coordinates. The setup is numerically efficient and does not rely on any positioning and orientation system (POS) based e.g. on an inertial measurement unit (IMU) or global navigation satellite system (GNSS) or wheel rotation counter on the autonomous vehicle.**

Keywords—object recognition; collision avoidance; autonomous vehicles; forestry.

I. Introduction

The quality of a situational model around an autonomous vehicle in natural environment is a difficult but important problem. This paper proposes a procedure to detect tree stems from individual light detection and ranging (LiDAR) scans. The tree stem lines are then used for an efficient LiDAR point cloud (PC) alignment. Finally, an environmental model including tree locations, sizes and ground details is being produced. The detection of tree locations around forest machines can be based on producing a situational model based on on-site terrestrial LiDAR data collected during harvesting operation. Accurate tree detection is necessary when the forest machines are being automated and the results of regenerate cuts and clear cuts are optimized.

Many of the previous works in this field focus on sparse distribution of trees and rely on high density point clouds with little noise produced through long duration scans from static platforms. Some of the literature focuses on producing visual models which do not necessarily correspond to the real world ground truth. When the data is excessively noisy or missing, artificial synthesis or extrapolation is commonly used as imputation method. When the tree is constructed, it is often assumed that points that are closer to each other are likely be on the same tree [1], [2], [3].

Iterative closest point (ICT) [10] is a classical method, which is typically used to match point clouds directly. The algorithm has several versions with increasing level of optimization. By denoting the PC size by n, typical complexities are $O(n\log(n))$ and $O(n)$ depending on the space partitioning details etc. An advantage can be gained, if the number n can be reduced e.g. by using the tree stem lines instead of points itself.

In our method we acquire the LiDAR signal as the harvester moves in on-site forest conditions. We get hundreds of point clouds from the LiDAR scans and align them into a universal coordinate system based only on the LiDAR information.

The central processing steps in this method are: 1) planarization of the point cloud (PC); 2) cutting the PC data to a layer with the local height z in between 0.5 m ≤ z ≤ 4.0 m; 3) a heuristical tree detection and 4) alignment of the consequent LiDAR scans into unified global coordinates based on step 3) information. The first novelty is in using a spatial angle filtering (SAF) [4] to produce the triangularized ground model in order to planarize the point cloud in the step 1). The second novelty is in using the tree stem centerlines to align the LiDAR scans in the step 4).

II. Data Acquisition

For testing the developed approach a total of 20 test areas in Vieremä, Finland were measured. For each area, a Velodyne 32-E LiDAR was mounted on a moving harvester. LiDAR technology utilizes pulsed laser light to measure the distance to the target. When the harvester was operating, the LiDAR scanned 180 degrees in horizontal and 41.3 degrees in vertical direction and generated a point cloud containing (x,y,z) points and their intensities. The coordinates are relative to the LiDAR position and the intensity reflects the strength of the laser return. The device is able to reach up to 70 m distance.

While generating the point cloud, the harvester moved along a trail. A total of 20 test areas were measured with varying harvester movements between 10 to 15 m producing an average of 600 scans, giving the distance between scans a few centimeters. Each single scan has about 40 000 points.

III. Data Processing

The point cloud was planarized by the SAF method [4], which produces a triangularized irregular network (TIN) model for the ground by filtering out sharp pikes and depressions at the triangles. The height difference between the original points and their projection at the TIN was then assigned as the height of the new points. The resulting point cloud was cut at heights +0.5 m and +4.0 m to eliminate the ground vegetation and the canopy. See Fig. 1 where a test area 16 is depicted after the planarization. The average ground triangle size is dependent of the point cloud density and the lengths of the triangle sides vary from 1.0 m at the center to 20 m at the perimeter of the area of interest.

A heuristic was used to select trees from the cut PC. A local point density ρ was computed on a regular grid with a δ= 0.2 m grid constant. Grids without points were assigned as $\rho 0$ =1/ρ_0^2 local point density. The average density of the original point cloud follows the inverse of the distance $r(p)$ of a point p from the scanning location (origo) and this needs to be taken into account in the method. The densities near the scanning origo are reduced by a term $r(p)/r_0$. Thus the local PC density $\varrho(p)$ is defined as:

$$\rho(p) = \frac{r(p)}{r_0} \frac{number\ of\ \ points\ at\ the\ grid\ square}{\delta^2} \qquad (1)$$

where r is the horizontal radius to the point p and r_0= 20 m is the distance at which the density is not reduced anymore.

To reduce the numerical problems caused by drastic fluctuations in density, the logarithmic of the density $log\ \varrho(p)$ will be used:

$$\log\rho(p) = \log(\max(\rho_0, \rho(p))) \ . \qquad (2)$$

The density $\log\rho(p)$ was then Gaussian filtered and localized by using the difference between the density and its average on a surrounding disk $disk(p)$ with radii r = 1.7,…,2.1 m.

The resulting relative logarithmic density $RLD(p)$ of a point p is:

$$RLD(p) = \log\rho(p) - \underset{q \in disk(p)}{\text{mean}} \log\rho(p) \qquad (3)$$

where q is another PC point.

The original log density log $\varrho(p)$ and its histogram can be seen on the left column of the Fig. 2 and the localized (relative) version $RLD(p)$ and the corresponding histogram are depicted on the right column.

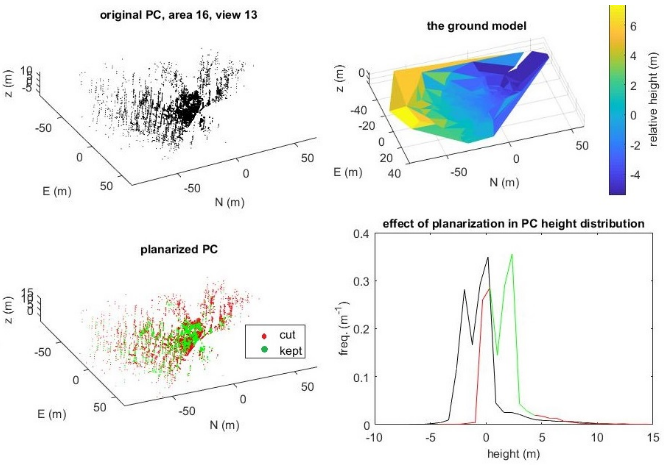

Fig. 1. Planarization of the PC. The original PC (black) of an individual view defines a TIN (above right), which is transformed to a planar one along the PC, see (lower left). Below right: effect of the vertical cut.

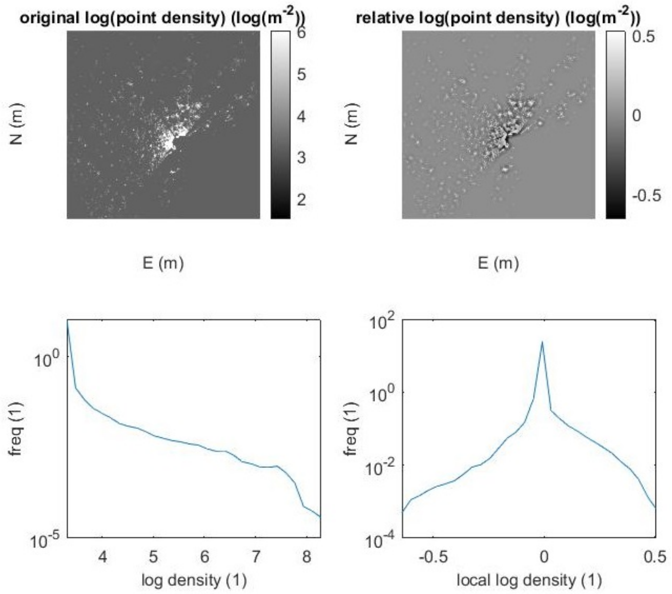

Fig. 2. *Left*: log of the point density of the cut PC (*above*) and the corresponding histogram (*below*). *Right*: local difference of the log density.and its histogram (*below*). This is the same test area as in Fig. 1.

The prominent tree locations with *RLD* maxima over a certain limit value *RLD > 0.4* were selected. If there were nearby trees, the tree with the higher *RLD* was chosen. The prominent tree locations of the test area of Fig. 1 are depicted in Fig. 3.

In order to align the consecutive LiDAR scans into a unified global coordinate system, the following procedure was used. For each scan, we selected 20 most likely tree positions identified by the heuristic method. The most likely trees are those with the highest *RLD* value. For each tree, a square of 25 cm in *xy*-plane around the tree position and all the points above the square in *z* direction were selected and called "region of interest" (*ROI*). Fig. 4 shows *ROI* around one tree in green. Zoomed illustration of the *ROI* of Fig. 4 is shown in Fig. 5. A line segment was fit to each *ROI* around a possible tree stem by using the largest eigenvalue λ_1 and the corresponding unit eigenvector v_1 of the covariance matrix C [5]:

$$C = \frac{1}{|ROI|} \sum_{p \in ROI} (p - p_0)(p - p_0)^T \tag{4}$$

where p_0 is the mean value of the $ROI \subset PC$.

The points on the fitted line segment were obtained from the following equation:

$$c(t) = p_0 + t v_1 \tag{5}$$

where t is the physical distance along the tree stem. Values of t are limited so that the center line is within the height cuts of the PC $0.5 \le e_3 \cdot c(t) \le 4.0\,(m)$.

A line segment fitted to *ROI* is shown in red color in Fig. 5. This phase serves also the sanity check for the heuristics producing locations for the potential trees depicted in Fig. 3. The heuristics of Eqs. 1–3 have been tuned to be overly optimistic producing approx. 20 % false positives. A false tree shows high s.t.d. of the horizontal radial distance $R(p)$ of the *ROI* from the red center line. The distance $R(p)$ is given by:

$$R(p) = \left\| \left(I - v_1 v_1^T\right)(p - p_0) \right\| \tag{6}$$

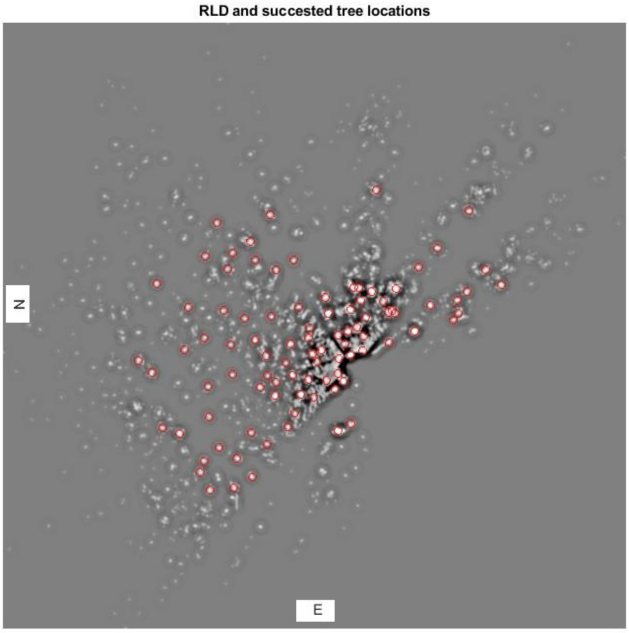

Fig. 3. Prominent tree locations in the test area depicted in Fig. 1.

where $I\text{-}v_1v_1^T$ is the projection matrix for a projection to the tree stem center line and ||.|| is the usual matrix norm. If the s.t.d. of $R(p)$, $p \in ROI$ is too large (e.g. 2 times the radius of a typical tree stem), then the *ROI* is a false positive and rejected.

Alignment continues with point clouds consisting only of the estimated line segments over all scans. First we defined two point clouds: reference point cloud (RPC) and moving point cloud (MPC), initially line segments from the first scan and second scan. We aligned the moving point cloud to the reference point cloud using the Matlab implementation of the Iterative Closest Point (ICP) algorithm [6]. In the first step of the ICP algorithm, each point in the moving point cloud PC_{i+1} is matched to the closest point in the reference point cloud PC_i producing a transformation T_i :

$$PC_{i+1} = T_i PC_i \,. \tag{8}$$

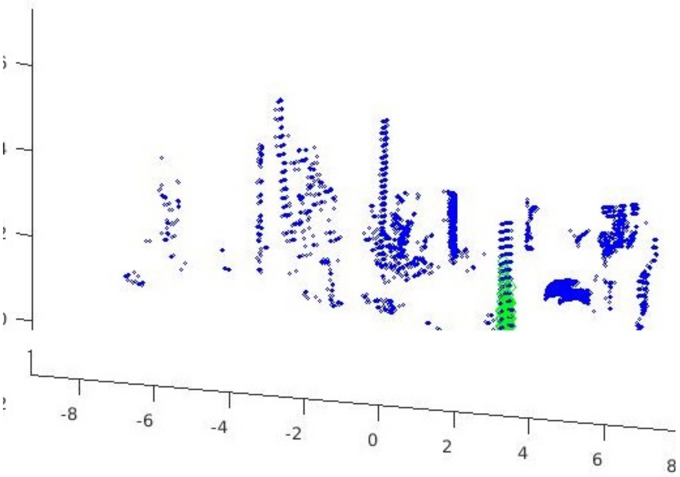

Fig. 4. *ROI* (green) around a tree.

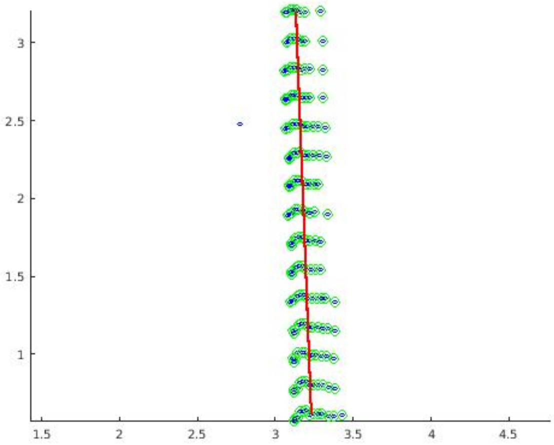

Fig. 5. Zoomed illustration of *ROI* (green) around a tree (blue). Line segment fitted to the *ROI*.

Since this process repeated over 600 views would produce a large error to the total transformation $T= T_nT_{n-1}...T_1$, the transformation (and PC alignment) were computed once for every 0.5 m movement. Then the rest of the views were added afterwards as sub tasks. The spatial step for the alignment has not yet been optimized for the accuracy nor for the computational efficiency.

Next the algorithm estimates the combination of rotation and translation which will best align each source point to its match found in the previous step. In this step we used the minimization of root mean square of point to point distance. These step are iterated until a convergence criteria is met. The algorithm stops when the average difference between estimated rigid transformations in the three most recent consecutive iterations falls below the following values: Euclidean distance between two translation vectors is less than 0.01 m and the angular difference in radians is less than 0.009.

The final obtained combination of rotation and translation was used to transform the moving cloud to the coordinate system defined by the reference point cloud. Then we set the moving point cloud as reference point cloud and set the line segments of the next scan as the moving point cloud. This continued until the last PC scan was reached.

IV. Results

Here we show results on a representative test area with pine trees. From this area the LiDAR measured altogether 608 scans, and each scan had approximately 20 000 points after planarization and usable PC extraction. Fig. 6 contains tree stem positions aligned from all consecutive scans shown with different colors. Line fitting is well suited for pine trees prevalent on the site.

V. Conclusions

This paper has proposed a new tree detection and point cloud alignment method based on on-site terrestrial LiDAR data collected during harvesting operation. Because the proposed method does not rely on any positioning and orientation system (POS) based e.g. on an inertial measurement unit (IMU) or global navigation satellite system (GNSS) or wheel rotation counter on the harvester, it is not sensitive to measurement errors of the devices.

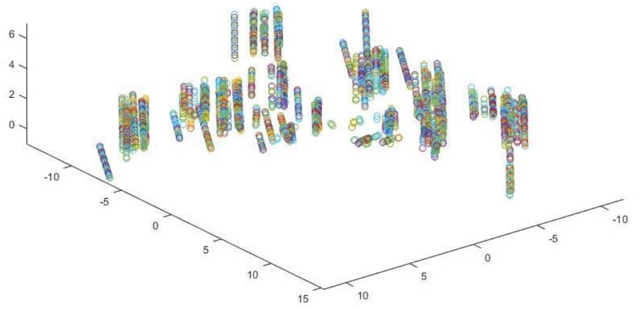

Fig. 6. Line fit for trees combined from 608 LiDARscans.

By zooming some of the trees of Fig.6 good performance is confirmed by observing that the aligned point clouds form smooth manifolds around the detected trees.

The estimation of the forest qualities have been based on aerial and terrestrial [7] PCs. The LiDAR scanners mounted to a harvester are rare, see eg. [8]. This work aims at optimal cutting decisions and developing a situational awareness of the harvester in order to enable further autonomous harvesters. A further study will be about the online tree stem profile and volume estimation directly from the LiDAR input. In this publications like [9] help to estimate the stem radius on different heights from the final aligned PC.

Acknowledgment

This research was supported by Business Finland grand (Dnro 2040/31/2017) related to the “New 3D analytics methods for intelligent ships and machines” project.

References

[1] H. Xu, N. Gossett, and B. Chen, “Knowledge and heuristic-based modeling of laser-scanned trees,” ACM Transactions on Graphics (TOG), vol, 26(4), 2007.

[2] Y. Livny, et al., “Automatic reconstruction of tree skeletal structures from point clouds,” ACM Transactions on Graphics (TOG), vol. 29(6): 151, 2010.

[3] DM. Yan, et al., “Efficient and robust reconstruction of botanical branching structure from laser scanned points,” in Proc. of 11th IEEE International Conference on Computer-Aided Design and Computer Graphics, pp 572–575, 2009.

[4] P. Nevalainen, M. Middleton, R. Sutinen, J. Heikkonen, and T. Pahikkala, ”Detecting Terrain Stoniness From Airborne Laser Scanning Data,” Remote Sens, vol. 8(9), 720, 2016. DOI: 10.3390/rs8090720

[5] [Online]. Available:
http://pointclouds.org/documentation/tutorials/normal_estimation.php

[6] Y. Chen and G. Medioni, “Object Modelling by Registration of Multiple Range Images. Image Vision Computing,” Butterworth-Heinemann, . vol. 10, pp. 145-155, 1992.

[7] B. Yang, W. Dai, Z. Dong, and Y. Liu, “Automatic Forest Mapping at Individual Tree Levels from Terrestrial Laser Scanning Point Clouds with a Hierarchical Minimum Cut Method, ” Remote Sensing, vol. 8(5), 372, 2016. DOI: 10.3390/rs8050372

[8] A. Salmivaara, et al. “Wheel rut measurements by forest machine-mounted LiDARsensors – accuracy and potential for operational applications,” International Journal of Forest Engineering, vol. 29, issue 1, 2018. DOI: 10.1080/14942119.2018.1419677

[9] A. Janowski, “The circle object detection with the use of Msplit estimation,” in Proc. of Seminary on Geomatics, Civil and Environmental Engineering (2017 BGC), vol. 26, 2018. DOI: 10.1051/e3sconf/20182600014

[10] Z. Zhang, “Iterative point matching for registration of free-form curves and surfaces,” International Journal of Computer Vision, vol. 13:2, pp. 119–152, 1994.