Object Detection and Terrain Classification in Agricultural Fields Using 3D Lidar Data

Mikkel Kragh(✉), Rasmus N. Jørgensen, and Henrik Pedersen

Department of Engineering, Aarhus University, Finlandsgade 22, Aarhus, Denmark
{mkha,rnj,hpe}@eng.au.dk

Abstract. Autonomous navigation and operation of agricultural vehicles is a challenging task due to the rather unstructured environment. An uneven terrain consisting of ground and vegetation combined with the risk of non-traversable obstacles necessitates a strong focus on safety and reliability. This paper presents an object detection and terrain classification approach for classifying individual points from 3D point clouds acquired using single multi-beam lidar scans. Using a support vector machine (SVM) classifier, individual 3D points are categorized as either ground, vegetation, or object based on features extracted from local neighborhoods. Experiments performed at a local working farm show that the proposed method has a combined classification accuracy of 91.6 %, detecting points belonging to objects such as humans, animals, cars, and buildings with 81.1 % accuracy, while classifying vegetation with an accuracy of 97.5 %.

Keywords: Object detection · Terrain classification · Agriculture · Lidar

1 Introduction

Autonomous farming is the concept of automatic agricultural machines operating safely and efficiently without human intervention. In order to ensure safe autonomous operation, robust real-time risk detection is crucial. Humans, animals, trees, other machines, etc. must be detected in due time to perform risk avoidance.

A lidar sensor measures range data to a set of surrounding points and generates a point cloud where each point is represented by a 3D position. It provides very accurate depth information in 360 ° horizontally and is robust towards changing lighting conditions. The lidar sensor has been used extensively in the automotive industry for detecting and localizing objects in urban environments by distinguishing between ground and obstacles [11]. In agriculture, however, a subdivision between objects and vegetation is necessary, since some apparent obstacles actually represent traversable crops. Therefore, a classification of points into ground, vegetation, and objects is needed. The ground class identifies accessible terrain, whereas the object class identifies obstacles/risks. The vegetation class serves as an intermediate category identifying both crops, bushes,

L. Nalpantidis et al. (Eds.): ICVS 2015, LNCS 9163, pp. 188–197, 2015.
DOI: 10.1007/978-3-319-20904-3_18

and trees. Depending on the agricultural context, vegetation can thus be either obstacles or a natural part of the field area.

In the literature, different approaches have been used to detect objects and characterize terrain in agricultural environments. [1,12–14] use single-beam lidar sensors and a mathematical density function for homogeneous grass to discriminate obstacles from grass and foliage. [6,15] use multi-beam lidars to perform ground plane identification in rough terrain. However, vegetation is not discriminated from objects. [8,9,18] use a feature-based approach for classifying individual points into the classes: scatter, linear, and surface. The objective is to identify vegetation (scatter); wires and tree branches (linear); and ground surfaces, rocks, and tree trunks (surface). [19] adds to this the objective of differentiating between vegetation and objects for increasing safety. This is done with a feature-based approach using online adaptation allowing the system to automatically collect and interpret training data. However, the results of this approach are only visually verified, and only a few specific cases are handled.

In this paper, we present an object detection approach for classifying individual points from 3D point clouds acquired with a vehicle-mounted Velodyne HDL-32E lidar. Our method calculates for each point 13 different features based on a local neighborhood. In order to account for the varying point density experienced with a vehicle-mounted lidar, we propose an adaptive neighborhood radius depending on the distance ensuring high resolution at short distance and preventing noisy features at far distance. Using a support vector machine (SVM), each point is categorized into one of three classes: ground, vegetation, or object.

The paper is divided into 5 sections. Section 2 presents the proposed approach including preprocessing, feature extraction, and classification. Section 3 presents the experimental setup and results followed by a discussion in Sect. 4. Ultimately, Sect. 5 presents a conclusion and future work.

2 Approach

The proposed method for object detection and terrain classification builds on individual point classification of single multi-beam lidar scans. A single lidar scan provides a 3D point cloud consisting of N points. For each point, 13 features are calculated using statistics from a local neighborhood. These features describe the distribution of points into surfaces, linear structures, clutter volumes, etc. and serve to distinguish between points representing the three classes: ground, vegetation, and object. Using hand labeled data, an SVM classifier is trained to classify individual points based on their calculated features.

2.1 Preprocessing

An initial step before extracting features performs a rotation and translation of the point cloud according to a globally estimated plane. This ensures that ground points in general lie close to the xy-plane. Due to variations in point density, the point cloud is first resampled using a minimum filter with a fixed sized radius

of 15 cm. A global plane is then estimated using a RANSAC-based plane fitting algorithm [5]. The point cloud is finally translated and rotated according to the normal vector of this plane. The resulting point cloud has an approximately vertically oriented z-axis.

2.2 Feature Extraction

When analyzing 3D data points from a point cloud, the notion of scale is extremely important in order to obtain both robust and accurate information. Point features are calculated using a local neighborhood such that the points located close to an evaluated point contribute with information of the point's context. For instance, one feature might describe how well a point fits with a local planar surface estimated on its neighborhood. The radius of the neighborhood should depend on the desired accuracy but also on the noise levels and the density of the point cloud. Depending on the sensor used for acquiring 3D data, a point cloud can be categorized as either dense or sparse [4]. A dense point cloud has an approximately constant point density, whereas the density of a sparse point cloud (e.g. from a single lidar scan) varies with the distance. Therefore, the process of feature extraction should incorporate information of the local point density and possibly also adjust the radius of the neighborhood accordingly.

Traditionally, the neighborhood radius is kept constant by dividing all points into a global voxel representation [7,8,19]. This approach allows for easy feature calculation and comparison since all voxels are the same size. However, it has the unfortunate property that it does not exploit the high point resolution close to the sensor, and at far distances only few measurements are available resulting in too noisy features. Different approaches have been made to handle this issue of varying point density. An automatic scale selection method estimates the optimal neighborhood radius that minimizes the error of local normal estimation [9]. Another approach is to perform feature extraction on multiple scales and choose the local scale that has the highest saliency [10,17]. However, these approaches both rely on a specific measure that cannot be generalized across all possible features and structures. Also, computing features at multiple scales significantly increases the computational complexity.

Therefore, in this paper we propose a simple heuristic approach that scales the neighborhood radius r linearly with the sensor distance d. This has the benefit of computational simplicity while allowing fine estimation close to the sensor and a more coarse estimate far from the sensor. The specific relationship is given as

$$r = 0.0276d + 0.25 \tag{1}$$

such that a radius of 0.3 m is used at a distance of 2 m, whereas a radius of 3.0 m is used at a distance of 100m.

It is important that all features are made scale-invariant such that the neighborhood radius does not directly influence the features. A common normalization technique is not applicable since the features express different characteristics. Hence, we need to consider normalization for each feature separately.

A total of 13 features related to the height, shape, orientation, distance, and reflectance are calculated. In the following, these are explained in detail, and individual normalization techniques are discussed.

f_1, f_2, f_3, f_4: Height. Four height related features are calculated inspired by the work in [15]. Height features capture structures that protrude from the ground either positively (upwards) or negatively (downwards). f_1 is simply the z-coordinate of the evaluated point i. f_2 is the minimum z-coordinate of the neighborhood. f_3 is the average z-coordinate of all points in the neighborhood. f_4 is the standard deviation of all z-coordinates. Since the standard deviation depends directly on the size of the neighborhood, it is normalized by dividing by the neighborhood radius r. In the following equations, z_i denotes the z-coordinate of the i'th point, and k denotes the number of points within a neighborhood of radius r. k thus varies with r and the specific point density locally around point i.

$$f_1 = z_i \tag{2}$$

$$f_2 = min\left(z_1 \dots z_k\right) \tag{3}$$

$$f_3 = \overline{z} = \frac{1}{k}\sum_{j=1}^{k} z_j \tag{4}$$

$$f_4 = \frac{\sigma_z}{r} = \frac{1}{r}\sqrt{\frac{1}{k}\sum_{j=1}^{k}\left(z_j - \overline{z}\right)^2} \tag{5}$$

f_5, f_6, f_7, f_8: Shape. Principal component analysis (PCA) of the point neighborhood can be used to describe the shape/saliency of the point cloud [8,18,19]. Let $\lambda_1 < \lambda_2 < \lambda_3$ be the eigenvalues of the 3×3 covariance matrix. In case of scattered points (random point distribution), $\lambda_1 \approx \lambda_2 \approx \lambda_3$. For points on planes, $\lambda_2, \lambda_3 \gg \lambda_1$, whereas for linear structures $\lambda_3 \gg \lambda_1, \lambda_2$. Using this intuition, λ_1 captures vegetation, $\lambda_2 - \lambda_1$ captures linear structures, whereas $\lambda_3 - \lambda_2$ captures planar-like data.

Constructing scale-invariant PCA features can be done in different ways. [10] scales λ_2 and λ_3 by the neighborhood radius but leaves λ_1 intact. This results in scale-invariant eigenvalues for planar-like data, whereas scatteredness is left unscaled. [16], on the other hand, uses the ratio of PCA values.

In this paper, we utilize the eigenvalue differences as described above and scale them by the largest eigenvalue. This guarantees scale-invariant features (always adds up to 1) while allowing for the differentiation between scatter, linear, and planar structures.

$$f_5 = \frac{\lambda_1}{\lambda_3} \tag{6}$$

$$f_6 = \frac{\lambda_2 - \lambda_1}{\lambda_3} \tag{7}$$

$$f_7 = \frac{\lambda_3 - \lambda_2}{\lambda_3} \tag{8}$$

In addition to the three PCA shape features, we use a normalized orthogonal residual sum of squares (RSS) proposed by [15].

$$f_8 = \frac{1}{k} \sum_{j=1}^{k} \left((\boldsymbol{p}_j - \overline{\boldsymbol{p}}) \cdot \boldsymbol{v}_1 \right)^2 \tag{9}$$

where $\boldsymbol{v}_1$ is the eigenvector corresponding to the smallest eigenvalue λ_1, $\boldsymbol{p}_i$ is the 3D vector of the i'th point, and $\overline{\boldsymbol{p}}$ is the neighborhood mean (centroid).

f_9, f_10, f_11: Orientation. From the principal component analysis, the eigenvector $\boldsymbol{v}_1$ is equal to the normal vector of a locally estimated plane. $\boldsymbol{v}_1$ thus describes the orientation of the plane. The z-component of the vector has been used to capture ground points assuming that the terrain is fairly flat and not sloped [10,15]. In this paper we include all the components.

$$f_9 = \boldsymbol{v}_1 \cdot (1, 0, 0) \tag{10}$$

$$f_{10} = \boldsymbol{v}_1 \cdot (0, 1, 0) \tag{11}$$

$$f_{11} = \boldsymbol{v}_1 \cdot (0, 0, 1) \tag{12}$$

f_12: Distance. Although the distance-dependent point density to some degree is handled by the varying neighborhood radius, the distance from a point $\boldsymbol{p}_j$ to the sensor $\boldsymbol{s}$ can also be used as a predictor [19].

$$f_{12} = \sqrt{(\boldsymbol{p}_i - \boldsymbol{s}) \cdot (\boldsymbol{p}_i - \boldsymbol{s})} \tag{13}$$

f_13: Reflectance. The lidar sensor utilized in the experiments provides for each point a reflectance intensity. This can help differentiate between different materials, although it depends also on the distance and incident angle [10,19].

$$f_{13} = \text{intensity}_i \tag{14}$$

2.3 Classification

A support vector machine (SVM) classifier is trained on hand-labeled data and used to differentiate between ground, vegetation, and object. In order to balance the training data, a number of ground and vegetation points, corresponding to

the number of object points, are drawn by random. We use the LIBSVM implementation [2] with a radial basis function (RBF) kernel and default SVM parameters $C = 1$ and $\gamma = \frac{1}{\#features} = \frac{1}{13}$. Prior to feeding the classifier, features are normalized by subtracting the mean and dividing by the standard deviation for each dimension across the training data. The normalization parameters are then stored for subsequent use in the test procedure.

3 Experiments and Results

An experimental dataset was acquired on a local working farm in Denmark in November 2014. Figure 1 shows the custom-built vehicle-mounted sensor platform including a Velodyne HDL-32E lidar [3]. In addition to the lidar sensor, a number of visual and pose sensors were mounted for subsequent analysis. The recordings include high and low grass, a large number of trees, 2 buildings, 2 cars, 5 men, 7 children, and 2 dogs, all from different angles and distances. 15 lidar frames from 7 different trials (recordings) were subsequently hand labeled into the three classes: ground, vegetation, and object. Results have been obtained using leave-one-out cross-validation (with 7 folds corresponding to the different trials), thereby training on 6 and testing on a single fold at a time. Separating trials in the cross-correlation should prevent overfitting, which would otherwise occur due to high correlation between frames within the same trial.

Table 1 presents a confusion matrix showing the accumulated counts of points across the 7 folds classified correctly or incorrectly compared to the ground truth. As mentioned above, the uneven distribution of ground, vegetation, and object points is evened out by drawing by random a number of these, corresponding to the number of object points, from individual frames. The results show a combined classification accuracy of 91.6 %. Points belonging to the ground are correctly predicted as ground with 96.4 % accuracy, and points belonging to vegetation are correctly predicted as vegetation with 97.5 % accuracy. Object points, however, are more often mistaken for vegetation, resulting in an object detection accuracy of 81.1 %.

Fig. 1. Sensor platform mounted on tractor.

Table 1. Confusion matrix relating predictions (columns) to ground truth (rows).

	Ground	Vegetation	Object
Ground	44806 (96.4 %)	1234 (2.7 %)	437 (0.9 %)
Vegetation	724 (1.6 %)	43372 (97.5 %)	381 (0.9 %)
Object	728 (1.6 %)	8041 (17.3 %)	37708 (81.1 %)

Figure 2 illustrates examples of two frames with ground truth labels and classifier predictions. The problem of object/vegetation confusion is particularly visible in Fig. 2b on the side of the building. Here, around half of the building is incorrectly predicted as vegetation.

Two feature selection techniques were used to investigate the individual importance of the 13 features. Both techniques use only a subset of all combinations of features, since exhaustive search is impractical with $\sum_{f=1}^{13} \binom{13}{f} = 8191$ combinations. In order to evaluate a feature combination, a common metric is needed. Since the features are ultimately used for classification, a wrapper method detecting possible interactions between features was used. The SVM classifier was thus trained on each feature combination, and the accuracy was used as a score.

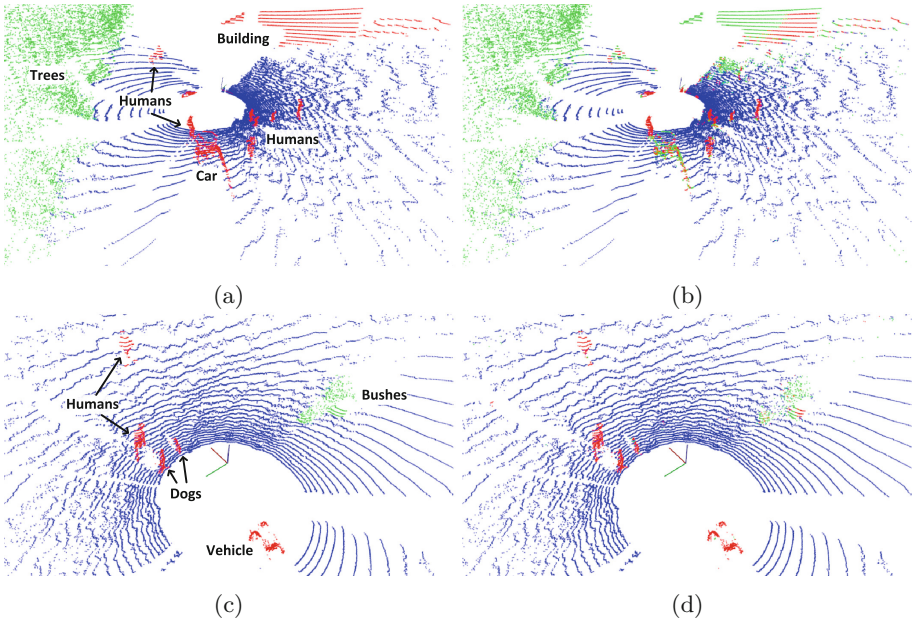
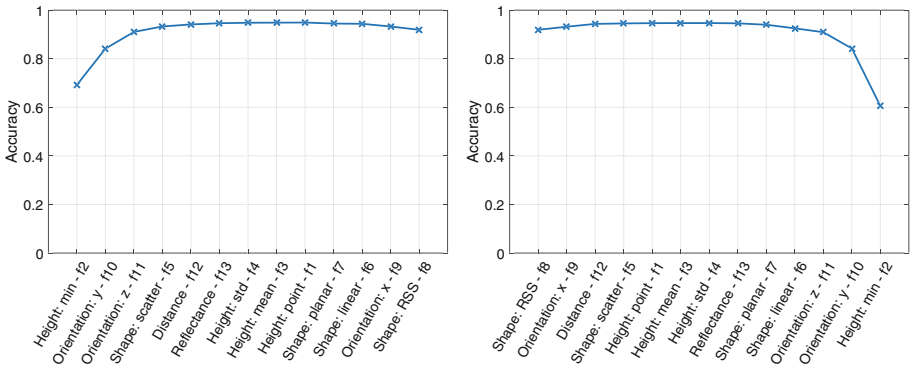


Fig. 2. Examples of classification results. a) and b) respectively show ground truth and classification results of a scene with ground, trees, humans, a car, and a building. c) and d) respectively show ground truth and classification results of a scene with ground, bushes, humans, and dogs. Blue denotes ground, green denotes vegetation, and red denotes objects (Colour figure online).



(a) Forward selection. (b) Backward elimination.

Fig. 3. Feature selection using greedy forward selection and backward elimination.

Greedy forward selection starts by evaluating all features individually and assigns for each a classification score. The feature with the highest score is added to a set of used features, and this set is gradually increased by iteratively adding the highest scoring feature of the remaining unused features. Figure 3a shows the relevance sorting of this approach. The most relevant feature is considered to be f_2 (minimum height), whereas the least relevant is f_8 (RSS).

Greedy backward elimination, on the other hand, starts by evaluating all features in combination leaving out a single feature. The feature that gives the smallest decrease in score is then eliminated, and the process is continued iteratively until a single feature is left. Figure 3b shows the relevance sorting of this approach. As for the forward selection, the most relevant feature is considered to be f_2 (minimum height), and the least relevant is f_8 (RSS).

All computations were performed using C++ on a laptop with an Intel i7 Quad-core CPU at 2.7GHz and 16GB of RAM. The average execution time is 705ms per frame. Preprocessing takes 2.4ms, feature extraction takes 324.9ms, and classification takes 377.9ms.

4 Discussion

Due to the interaction of features, the two feature selection techniques do not fully agree about the sorting of all relevances. However, some observations can be made from the graphs. Using more than 5 features seems to be unnecessary, as it does not significantly increase the accuracy. This is an important observation, since utilizing fewer features results in decreased computational complexity. Another common trend of the two graphs is seen by looking at the three feature categories: height, shape, and orientation. Only one or two features within each category are considered relevant. This implies (but does not prove) that features within each of the categories are correlated and thus redundant. Although the two techniques do not agree about the specific features, they both include a

height, shape, and orientation feature among the most relevant four features. A reasonable choice of feature reduction would therefore be to select the intersection of the 5 most significant features from the two selection techniques.

5 Conclusion

In this paper, we have presented an object detection approach for classifying individual points from 3D point clouds acquired with a vehicle-mounted lidar. Our method calculates for each point 13 different features based on a local neighborhood. In order to account for the varying point density experienced with a vehicle-mounted lidar, the neighborhood radius depends on the distance ensuring high resolution at short distance and preventing noisy features at far distance. Using a support vector machine, each point is categorized into one of three classes: ground, vegetation, or object.

The proposed method shows promising results on an experimental dataset recorded on a working farm including grass, trees, buildings, cars, humans, and animals. It has a combined classification accuracy of 91.6 %. Ground points are correctly classified with an accuracy of 96.4 %, and points belonging to vegetation are correctly predicted as vegetation with 97.5 % accuracy. Object points, however, are more often mistaken for vegetation, resulting in an object detection accuracy of 81.1 %.

In order to increase differentiation performance, further work will focus on temporal accumulation of lidar frames using odometry information from GPS and IMU sensors. Also, further differentiation and characterization of objects will require additional information possibly by fusing lidar and vision sensors.

Acknowledgements. This research is sponsored by the Innovation Fund Denmark as part of the project "SAFE - Safer Autonomous Farming Equipment" (project no. 16-2014-0).

References

1. Castano, A., Matthies, L.: Foliage discrimination using a rotating ladar. In: 2003 IEEE International Conference on Robotics and Automation, vol. 1, pp. 1–6 (2003)
2. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(3), 1–27 (2011). Article No. 27
3. Christiansen, P., Kragh, M., Steen, K.A., Karstoft, H., Jørgensen, R.N.: Advanced sensor platform for human detection and protection in autonomous farming. In: 10th European Conference on Precision Agriculture (ECPA 2015) (2015)
4. Douillard, B., Underwood, J., Kuntz, N., Vlaskine, V., Quadros, A., Morton, P., Frenkel, A.: On the segmentation of 3D lidar point clouds. In: Proceedings - IEEE International Conference on Robotics and Automation, pp. 2798–2805. IEEE (2011)
5. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)

6. Hadsell, R., Bagnell, J.A., Huber, D., Hebert, M.: Space-carving kernels for accurate rough terrain estimation. Int. J. Robot. Res. **29**(8), 981–996 (2010)
7. Hebert, M., Vandapel, N.: Terrain classification techniques from ladar data for autonomous navigation. In: Collaborative Technology Alliances Conference (2003)
8. Lalonde, J.F., Vandapel, N., Huber, D.F., Hebert, M.: Natural terrain classification using three-dimensional ladar data for ground robot mobility. J. Field Robot. **23**(10), 839–861 (2006)
9. Lalonde, J.F., Unnikrishnan, R., Vandapel, N., Hebert, M.: Scale selection for classification of point-sampled 3D surfaces. In: Proceedings of International Conference on 3-D Digital Imaging and Modeling, 3DIM, pp. 285–292 (2005)
10. Lim, E.H., Suter, D.: 3D terrestrial LIDAR classifications with super-voxels and multi-scale conditional random fields. CAD Comput. Aided Des. **41**(10), 701–710 (2009)
11. Luettel, T., Himmelsbach, M., Wuensche, H.J.: Autonomous ground vehiclesconcepts and a path to the future. In: Proceedings of the IEEE, vol. 100, (Special Centennial Issue), pp. 1831–1839, May 2012
12. Macedo, J., Manduchi, R., Matthies, L.: Ladar-based discrimination of grass from obstacles for autonomous navigation. In: Proceedings of the International Symposium on Experimental Robotics VII (ISER 2001), pp. 111–120 (2001)
13. Manduchi, R., Castano, A., Talukder, A., Matthies, L.: Obstacle detection and terrain classification for autonomous off-road navigation. Auton. Robots **18**(1), 81–102 (2005)
14. Matthies, L., Bergh, C., Castano, A., Macedo, J., Manduchi, R.: Obstacle detection in foliage with ladar and radar. In: Proceedings of ISRR, pp. 291–300 (2003)
15. McDaniel, M.W., Nishihata, T., Brooks, C.A., Lagnemma, K.: Ground plane identification using LIDAR in forested environments. In: Proceedings - IEEE International Conference on Robotics and Automation, pp. 3831–3836 (2010)
16. Spinello, L., Arras, K.O., Triebel, R., Siegwart, R.: A layered approach to people detection in 3d range data. In: Proceedings of the AAAI Conference on Artificial Intelligence: Physically Grounded AI Track (AAAI) (2010)
17. Unnikrishnan, R., Hebert, M.: Multi-scale interest regions from unorganized point clouds. In: 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops (2008)
18. Vandapel, N., Huber, D., Kapuria, A., Hebert, M.: Natural terrain classification using 3-D ladar data. In: Proceedings of IEEE International Conference on Robotics and Automation, ICRA 2004, vol. 5, pp. 5117–5122 (2004)
19. Wellington, C., Stentz, A.: Online adaptive rough-terrain navigation in vegetation. In: Proceedings of IEEE International Conference on Robotics and Automation, ICRA 2004, vol. 1, pp. 96–101 (2004)