

Lidar-based detection of airborne particles for robust robot perception

Leo Stanislas, Niko Sünderhauf, Thierry Peynot
Queensland University of Technology, Brisbane, Australia
email: l.stanislas@qut.edu.au

Abstract

Airborne particles such as dust, smoke and fog have a significant detrimental impact on Lidar-based robotic perception systems. Lidar rays can reflect on these particles, leading modern perception methods to erroneous results, such as false obstacles or misclassified elements. We propose a method to detect airborne particles in 3D Lidar point clouds using classification from geometric features and Lidar intensity returns. We compare three different classifiers and evaluate our approach on dust and fog data collected in outdoor scenarios. We achieve an accuracy of up to 95% in detecting airborne particles in Lidar point clouds, making our proposed method a promising solution for applications such as obstacle detection and object recognition in outdoor environments. We make available the code and data for this work at https://leo-stan.github.io/particles_detection.

1 Introduction

Perception is a key factor for enabling robots to understand the world and perform appropriate actions. In outdoor scenarios, perception algorithms are challenged by many environmental parameters, one of which is the presence of airborne particles such as smoke, dust, or fog [Peynot *et al.*, 2010]. Such particles are observed in the data produced by sensors commonly used in robotics such as Lidars [Phillips *et al.*, 2017]. Lidar points corresponding to airborne particles can lead to information being misinterpreted by perception algorithms (e.g false obstacles and misclassification). For instance, a dust cloud led Boss, the winner of the DARPA Urban Challenge, to a false stop during the competition because it interpreted the dust as an obstacle [Urmson *et al.*, 2008]. Existing research conducted to mitigate the detrimental effects of airborne particles on Lidar-based perception systems follows two main approaches.

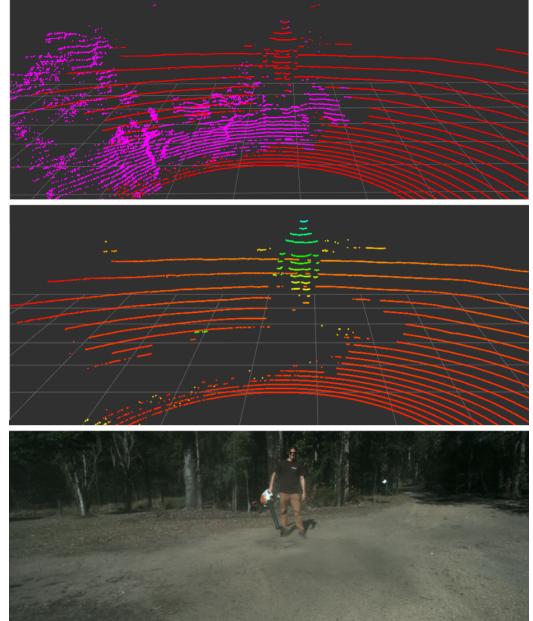


Figure 1: *Top*: Detection of airborne dust particles in a Lidar point cloud, purple is Particle and red is Non-Particle. *Middle*: Removal of detected dust particles from point cloud for robust perception, the colour is mapped to the vertical axis (red is low, green is high). *Bottom*: Image depicting the scene.

One approach relies on the fact that multiple types of sensors (e.g. Lidar, Radar, Camera, etc) react differently to the presence of airborne particles. Thus, sensor data affected by airborne particles can be identified by detecting inconsistencies between data from different sensor types [Peynot and Kassir, 2010; Gerardo-Castro *et al.*, 2013]. However, this approach requires multiple types of sensors in a single perception system and relies on assumptions about the behaviour of the different types of sensors in the presence of airborne particles. For instance, assuming that a Radar sensor will not detect dust particles but a Lidar sensor will.

Another approach seen in the literature is based on the classification of data corresponding to airborne particles [Vidal-Calleja and Agammenoni, 2012]. Once detected, this data can be handled appropriately depending on a particular perception application. For instance, Lidar points corresponding to dust particles could be discarded to avoid generating false obstacles, or smoke detected in a camera image could lead to trigger a smoke alarm. However, this approach has only been applied to vision-based perception systems.

We propose a method to detect airborne particles based on data from a 3D Lidar sensor. We use a classification method based on appearance and geometric features from the 3D Lidar point clouds to identify Lidar points as either particles or non-particles. We also provide a new dataset with 3D Lidar and stereo camera sensor data in natural scenes with dust and fog particles. Our dataset contains more than 4,500,000 labelled Lidar points in five different scenes. Fig. 1 shows the successful classification of dust particulate in a Lidar point cloud in a scene with a human and dust. The ability to differentiate between Lidar points that correspond to airborne particles and those that do not is a step forward to prevent the generation of false obstacles and the misclassification of objects or terrains in challenging outdoor conditions.

This article is structured as follows; Section 2 discusses related work in robot perception in the presence of airborne particles, voxel maps, and point cloud classification. Section 3 presents our airborne particle detection method. Section 4 provides an experimental evaluation of our approach on real Lidar data in outdoor scenarios. Finally, Section 5 draws conclusions and discusses future research.

2 Related Work

A comprehensive study of the impact of dust particles on Lidar data was performed in [Phillips *et al.*, 2017]. They analysed the impact of different thicknesses of dust clouds for different distances to the sensor. They also provide qualitative results for Lidar point clouds impacted by dust in mining applications. They concluded that dust particles have a systematic effect on Lidar measurements.

Multiple approaches have been explored to mitigate the adverse impact of airborne particles in Lidar point clouds. Gerardo-Castro *et al.* developed a point-to-point comparative test that they called consistency test, to identify discrepancies between Lidar and Radar points [Gerardo-Castro *et al.*, 2013] taking advantage of the fact that Radar data is not impacted by airborne particles such as dust and smoke. When a discrepancy between a Lidar and Radar point was found, they disregarded the Lidar point assuming it was generated from

hitting dust particles. They later developed their consistency test with Gaussian Processes to estimate continuous surfaces between sensor points [Gerardo-Castro *et al.*, 2014; Gerardo-Castro *et al.*, 2015]. A similar method proposed by [Peynot and Kassir, 2010] used inconsistency in data between a Lidar point cloud and a colour image to identify Lidar points corresponding to dust particles. Whenever the range gradient of the point cloud was different to the intensity gradient in the image, the point cloud was identified as being generated by an airborne particle. Separately, Vidal Calleja *et al.* proposed a vision-based smoke detection system based on image classification [Vidal-Calleja and Agammenoni, 2012]. They classified visual features as smoke or non-smoke using a bag of words paradigm based on Latent Dirichlet Allocation (LDA). They achieved an accuracy of 89.61% over 151 images. However, to the best of the author’s knowledge, this classification approach has not been applied to 3D Lidar data. Our method is based on data from a single 3D Lidar. We use a classifier to detect Lidar points generated by airborne particles based on appearance and geometric information contained in the Lidar point cloud.

Lidar point cloud classification is a common process, especially for applications such as terrain classification. [Lalonde *et al.*, 2006] used geometrical features obtained from a Principal Component Analysis (PCA) to perform classification of the ground with a Gaussian Mixture Model (GMM). [Wurm *et al.*, 2009] introduced the use of Lidar intensity returns as a feature to recognise vegetation from road surfaces with a Support Vector Machine (SVM) classifier. [Suger *et al.*, 2016] made use of both geometrical features and intensity returns to classify up to four terrain types with a Random Forest (RF) classifier. Recently, deep neural networks have been introduced to perform object detection (e.g. cars, pedestrians, bicycles) in Lidar point clouds [Engelcke *et al.*, 2016; Zhou and Tuzel, 2017; Chen *et al.*, 2016]. Such approaches learn features directly from raw Lidar data as opposed to hand-crafted features, therefore, they require a much larger number of labelled data to be trained than classifiers such as SVM or RF. In this work, we evaluate both SVM and RF classifiers with hand-crafted features based on geometric and intensity information from Lidar points. We also implement a Neural Network classifier with features learnt directly from raw Lidar data.

A number of datasets containing Lidar data are publicly available such as the popular KITTI dataset [Geiger *et al.*, 2012] or the MIT DARPA Urban Challenge [Leonard *et al.*, 2007]. The Marulan dataset [Peynot *et al.*, 2010] provides Lidar data with airborne particles such as dust and smoke. However, no intensity information is provided, and only 2D Lidar sensors were used in the experiments. Thus, we recorded a

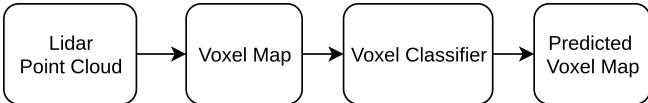


Figure 2: Airborne particles detection method.

new dataset containing dust and fog particles using a 3D Velodyne HDL-32E Lidar in multiple natural scenes to evaluate the approach proposed in this paper. We recorded both geometry and intensity information for the two echoes returned by the Lidar.

3 Airborne Particles Detection

Our detection method is a three-stage process; first, we discretise the raw Lidar point cloud into 3D voxels of fixed dimensions. Second, we format the data contained in each voxel for the voxel classifier. Third, we perform classification on each voxel and assign a label to that voxel. The voxels identified as containing airborne particles can then easily be handled appropriately for any specific application. A diagram of the method is presented in Fig. 2.

3.1 Voxel Map

Discretisation of a point cloud is common practice to store and process spatial sensor data. 2D grids [Elfes, 1989] and 3D voxel maps [Hornung *et al.*, 2013] are popular methods used in robotics to perform obstacle detection or classification from point clouds.

We discretise the Lidar scans by dividing the 3D space in equally spaced non-overlapping voxels of equal size $[v_X, v_Y, v_Z]$. The dimensions of this voxel map are $[m_X, m_Y, m_Z]$ and the map is centred around the Lidar’s origin. Lidar points are accumulated in their corresponding voxels based on their position in the map. Each voxel is defined by $V = \{p_i = [x_i, y_i, z_i, a_i] \in \mathbb{R}^4\}_{i=1 \dots n}$ with x_i, y_i, z_i position and a_i intensity of each Lidar point p_i , and n the number of Lidar points in the voxel. The x_i, y_i, z_i position of each Lidar point is relative to the origin of the map. The intensity a_i is a value in the range $[0; 100]$ that indicates the amount of light reflected by the surface hit by the Lidar point. This value is dependent on a number of factors such as the range, the angle of incidence of the light ray to the surface, and the material of the surface.

3.2 Voxel-wise Classification

We define two classes for our binary classifier, particle and non-particle, the latter corresponding to anything that is not considered an airborne particle. We investigate three classifiers, a Random Forest [Breiman, 2001], a Support Vector Machine (SVM) [Cortes and Vapnik, 1995], and a custom Neural Network architecture.

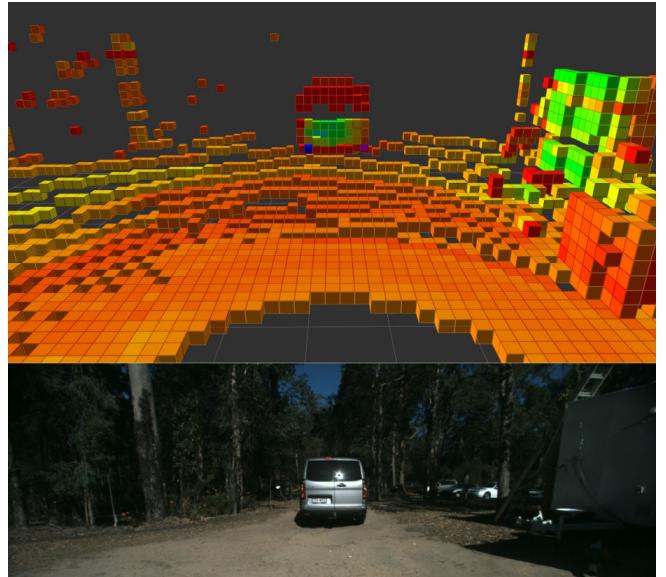


Figure 3: Top: Voxel map computed from a Lidar point cloud. Colour represents Lidar intensity, red is low, green is high. Bottom: Visual picture of the corresponding scene

Random Forest and SVM Classifiers

To use a Random Forest or SVM Classifier we compute a set of hand-crafted features from Lidar points. For each voxel, we define an input feature vector as $F_{in} = [\mu_a, \sigma_a, \rho, \varphi]$ computed from the Lidar points in that voxel. μ_a and σ_a are the Lidar intensity mean and standard deviation computed with standard statistical operations. These appearance features partly reflect the type of material that has been hit by the Lidar rays in the voxel. The *Roughness* ρ and *Slope* φ are geometrical features computed using a Principal Component Analysis of the 3D Lidar points. The Roughness is defined by [Suger *et al.*, 2016] as the smallest eigenvalue

$$\rho = \min_k \lambda_k, \quad (1)$$

with $\lambda_{[1:3]}$ the set of eigenvalues. The Roughness represents a measure of how organised the points are in the space. For instance, a smooth flat surface will have a low roughness value, while a scattered point cloud will have a high roughness value. The slope is defined by [Suger *et al.*, 2016] as the angle between the eigenvector with the lowest variance and the ground plane.

$$\varphi = \arcsin \alpha_{min}, \quad (2)$$

with α_{min} the eigenvector corresponding to the smallest eigenvalue. The Slope angle helps identify vertical surfaces from horizontal surfaces. As the ground plane, we assume a locally flat ground around the robot and use the xy plane of the voxel map. We chose

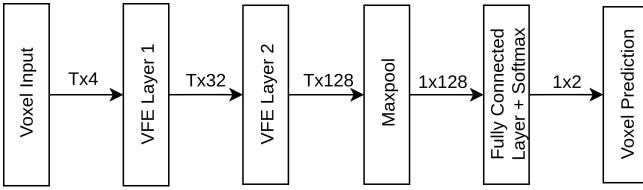


Figure 4: Neural network architecture used to detect airborne particles in this paper.

this set of feature for their success in other Lidar-based classification applications [Lalonde *et al.*, 2006; Suger *et al.*, 2016]. We use the input vector F_{in} computed for each voxel to predict the label of the voxel using either a Random Forest or SVM Classifier.

Neural Network Classifier

We investigate the ability of a Neural Network to detect airborne particles by learning appearance and geometrical features directly from raw Lidar data. To do so, a maximum of T Lidar points per voxels are randomly sampled to form a 2D input feature vector $F'_{in} = \{[a_i, x_i - \hat{x}, y_i - \hat{y}, z_i - \hat{z}]\}_{i=1\dots T}$, where a_i is the intensity return, and $\hat{x}, \hat{y}, \hat{z}$ is the average of the position of all points in the voxel. This is used to compute the relative position $x_i - \hat{x}, y_i - \hat{y}, z_i - \hat{z}$ of each individual Lidar Points. The intent is for the Neural Network to learn appearance features from the raw Lidar intensity values, and geometrical features from the relative position of the Lidar points within a voxel. The computed feature vector F'_{in} is inserted into our Neural Network for training and prediction.

Our Neural Network architecture (illustrated in Fig. 4) starts with two Voxel Feature Encoding (VFE) layers from [Zhou and Tuzel, 2017] to increase the dimensionality of the feature space from \mathbb{R}^4 to \mathbb{R}^{128} . The VFE layers (Fig. 5) are presented as $VFE(c_{in}, c_{out})$ where c_{in} and c_{out} are the input and output feature dimensions respectively. These layers encode point cloud features using a fully connected layer with output dimension $c_{out}/2$ concatenated with an element-wise Maxpool to produce a point-wise concatenated feature of dimension c_{out} . The two VFE layers are followed by another element-wise Maxpool layer and a last fully connected layer with output dimension 2 and Softmax to obtain a final vector with prediction scores. We use the Adam optimiser as our loss function [Kingma and Ba, 2014] and train our Neural Network from scratch.

4 Experiments

We recorded a custom dataset to evaluate our airborne particle detection method in multiple scenarios with various types of particles and obstacles.

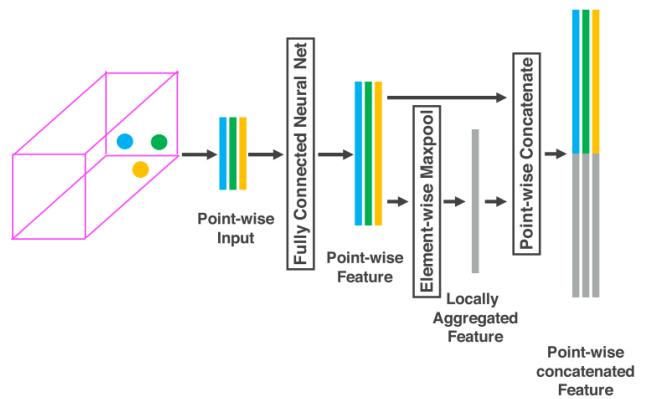


Figure 5: Voxel Feature Encoding (VFE) layer from [Zhou and Tuzel, 2017].



Figure 6: Our Clearpath Husky robot surrounded by dust particles during experiments.

4.1 Experimental Setup

To record the data and evaluate our detection method, we fitted a Velodyne HDL-32E Lidar sensor on a Clearpath Husky platform (shown in Fig. 6). We computed our experimental results using a standard desktop computer with a 3.4GHz Intel i7 CPU and a Nvidia GTX 1070 GPU. The dimensions of the voxel maps used during experiments were $m_X = m_Y = 20m$ and $m_Z = 3m$ centred around the robot. A voxel size of $v_X = v_Y = v_Z = 0.2m$ was chosen to have enough Lidar points in each voxel to carry geometrical information, while preserving sufficient details in the scene.

4.2 Airborne Particle Dataset

With no publicly available 3D Lidar dataset with airborne particles and intensity returns, we recorded an

extensive dataset to evaluate our airborne particle detection method. Two types of particles are recorded, fog from an Antari F80Z fog machine with heavy duty Antari fog liquid and dust using a leaf blower in dusty areas. Fig. 7 shows two scenes with airborne particles with a colour image and the corresponding Lidar point cloud. In Fig. 7a a dust cloud is generated on a flat ground surface, the impact of dust particles can be clearly seen in the Lidar point cloud with a large number of Lidar points above the ground plane in the centre of the scene. Similarly, Fig. 7b shows a scene with fog particles above flat ground. Again, the airborne fog particles can be observed in the Lidar point cloud with points above the ground. Our dataset contains more than 4,500,000 labelled Lidar points with 50% particle and 50% non-particle labels. We stored all the data produced by our Velodyne HDL-32E sensor, including position, intensity, and which of two echoes. We recorded five scenes in outdoor environments with various non-particle elements such as buildings, cars, humans, trees and foliage. We labelled our ground truth data directly from the raw Lidar points based on observations of the scene and prior knowledge about the experiments. We make this dataset publicly available with the code used in this work.

4.3 Classification Performance

To train our classifier, we randomly divided our Lidar points into three splits with 60% training, 20% validation, and 20% testing sets of data. We computed voxels from each set of Lidar points, with a voxel size of $0.2m$ we obtained roughly 300,000 voxels in our training set, and 100,000 in our validation and testing sets. Our SVM classifier used a Radial Basis Function (RBF) kernel with a penalty parameter of 1 and a kernel coefficient of 0.25. Our Random Forest classifier used 100 decision trees and the Gini index as the split criterion. These parameters were obtained using a random search over a range of values and a 5-fold cross-validation step. During training, our Neural Network classifier was set with a learning rate of 0.001, a batch size of 64 and trained for 25 epochs. During prediction, our GPU had enough memory to predict the entire voxel map in a single batch. These parameters were chosen empirically to obtain the best performance. We compute four different performance metrics. The accuracy of each classifier is calculated by

$$Accuracy = \frac{TP + TN}{N}, \quad (3)$$

with TP and TN the number of True Positives and True Negative predictions respectively and N the total number of predicted voxels. We compute the Precision and Recall scores as

$$Precision = \frac{TP}{TP + FP}, \quad (4)$$

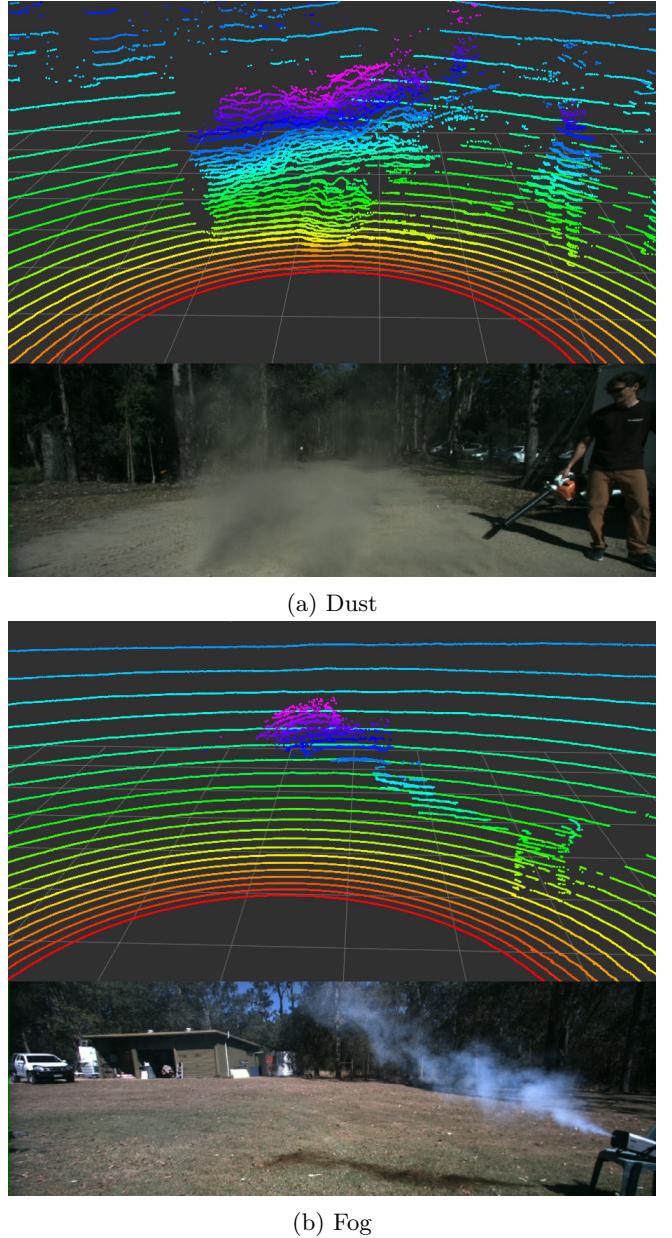


Figure 7: Impact of dust and fog in 3D Lidar point clouds for two natural scenes. Lidar point clouds are shown above a corresponding image of the scene. The colour of the point cloud corresponds to each of the 32 infrared light rays emitted by our Velodyne HDL-32E sensor.

Table 1: Comparison of classification performance between Random Forest and Neural Network classifiers.

Classifier	SVM	Random Forest	Neural Network
Accuracy	92%	93%	95%
Precision	0.89	0.91	0.94
Recall	0.96	0.96	0.97
F1 score	0.92	0.93	0.95
Prediction time in sec	1.800	0.020	0.001

Table 2: Confusion matrices for our Neural Network classifier.

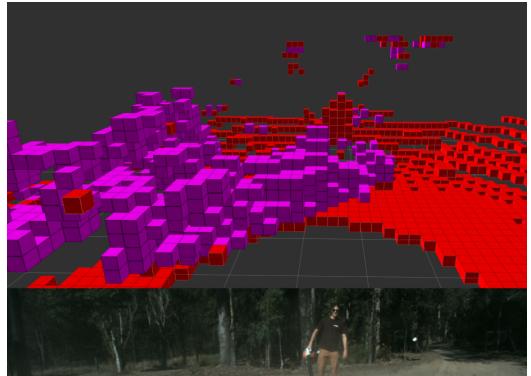
Class	Particle	Non-Particle
Particle	0.95	0.05
Non-Particle	0.03	0.97

$$Recall = \frac{TP}{TP + FN}, \quad (5)$$

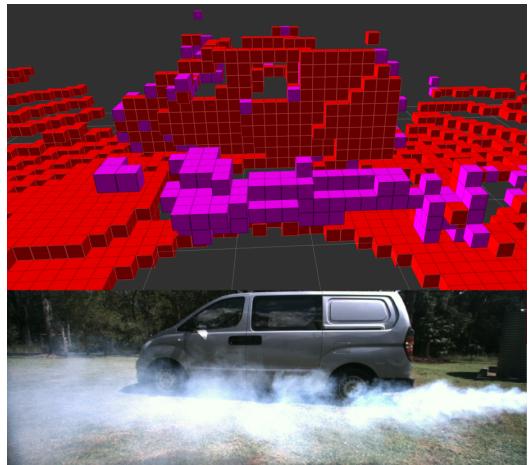
with FP and FN the number of False Positive and False Negative predictions. Finally, we compute the F1-score with

$$F1 = 2 \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

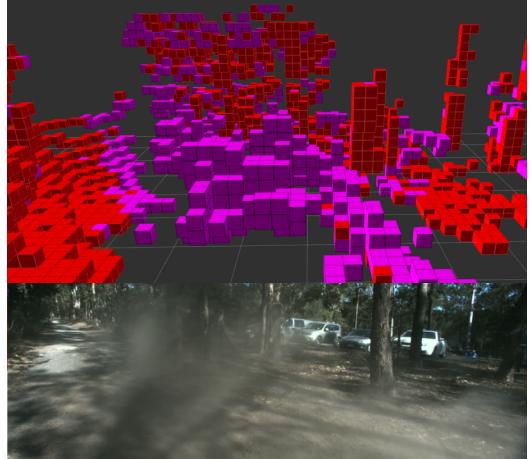
We present the classification performance quantitatively in Table 1. Our SVM and Random Forest classifiers achieved comparable performance with 92% and 93% accuracy respectively, while our Neural Network classifier achieved the best performance at 95% accuracy. We also compared the computation time of the prediction step of each classifier; the numbers presented are an average over 50 predictions. Our SVM and Random Forest classifiers were computed on a CPU with an average prediction time of 1.8s and 0.02s respectively. Our Neural Network prediction step was computed on a GPU, providing the fastest prediction time of 0.001s. Table 2 shows the confusion matrices for our Neural Network classifier. The true detection rate for the Particle class is less than the Non-Particle. These results show that our method is having more difficulties classifying correctly Particles than Non-Particles voxels. This is a positive result for applications such as obstacle detection where accurately detecting obstacles as Non-Particle is critical. Figure 8a shows qualitative results of scan predictions in different environments. From these results, we can see that our method successfully learnt to detect airborne particles of dust and fog. Moreover, when an obstacle is partly hidden behind particles (e.g. human in Fig. 8a, or tree trunks in Fig. 8c) it is correctly detected as Non-Particle. A minority of voxels can be seen as misclassified on the car in Fig. 8b most likely due to the geometry of the car and the interaction with the Lidar rays with the windows. While these results are conclusive, the perfor-



(a) Dust particles with a human in the scene.



(b) Fog with a car in the background



(c) Dust in natural environment

Figure 8: Label prediction of 3D voxels using our Neural Network classifier in the presence of dust and fog in Lidar point clouds. Purple voxels are predicted as particles while the red voxels are predicted as non-particles.

mance of the classifiers could be improved, for instance, by improving the false positive rate of airborne particles to guarantee that no obstacles would be considered as particles in critical obstacle avoidance applications.

5 Conclusion and Future Work

In this paper, we presented a Lidar-based method for the detection of airborne particles for robotic applications. We explored three different classifiers and compared their performance using real Lidar data of dust and fog particles. We found the best performance was achieved by our Neural Network classifier with 95% accuracy. This classifier also proved to be the fastest to perform label prediction.

In future work, we plan to explore the potential of the multi-echo return of our Lidar sensor as a classification feature. We also plan to perform data fusion with a stereo camera to use the best of both sensors in the detection process. Finally, we plan to test more types of particles such as smoke and raindrops with our robot platform performing real-time navigation.

Acknowledgements

The authors wish to acknowledge Leo Stanislas' PhD Scholarship from QUT, QUT's High-Performance Computing system, QUT's Samford Ecological Research Facility (SERF) where the experiments were conducted, and Marcus Yates for helping during the field trials.

References

- [Breiman, 2001] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [Chen *et al.*, 2016] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. *CoRR*, 2016.
- [Cortes and Vapnik, 1995] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [Elfes, 1989] Alberto Elfes. Using Occupancy Grids for Mobile Robot Perception and Navigation. *IEEE Computer*, 1989.
- [Engelcke *et al.*, 2016] Martin Engelcke, Dushyant Rao, Dominic Zeng Wang, Chi Hay Tong, and Ingmar Posner. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. *CoRR*, 2016.
- [Geiger *et al.*, 2012] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [Gerardo-Castro *et al.*, 2013] Marcos P. Gerardo-Castro, Thierry Peynot, and Fabio Ramos. Laser-radar data fusion with Gaussian process implicit surfaces. *Springer Tracts in Advanced Robotics*, 2013.
- [Gerardo-Castro *et al.*, 2014] Marcos P. Gerardo-Castro, Thierry Peynot, Fabio Ramos, and Robert Fitch. Robust Multiple-Sensing-Modality Data Fusion using Gaussian Process Implicit Surfaces. *IEEE International Conference on Information Fusion*, 2014.
- [Gerardo-Castro *et al.*, 2015] Marcos P. Gerardo-Castro, Thierry Peynot, Fabio Ramos, and Robert Fitch. Non-Parametric Consistency Test for Multiple-Sensing-Modality Data Fusion. *IEEE International Conference on Information Fusion*, 2015.
- [Hornung *et al.*, 2013] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013.
- [Kingma and Ba, 2014] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, 2014.
- [Lalonde *et al.*, 2006] Jean-François Lalonde, Nicolas Vandapel, Daniel F. Huber, and Martial Hebert. Natural Terrain Classification Using Three-Dimensional Lidar Data for Ground Robot Mobility. *Journal of Field Robotics*, 23(10), 2006.
- [Leonard *et al.*, 2007] John Leonard, J How, S Teller, M Berger, S Campbell, G Fiore, et al. Mit darpa urban grand challenge team public data, 2007.
- [Peynot and Kassir, 2010] Thierry Peynot and Abdallah Kassir. Laser-camera data discrepancies and reliable perception in outdoor robotics. *IEEE International Conference on Intelligent Robots and Systems*, 2010.
- [Peynot *et al.*, 2010] Thierry Peynot, Steve Scheding, and Sami Terho. The Marulan Data Sets:Multi-Sensor Perception in Natural Environment With Challenging Conditions. *The International Journal of Robotics Research*, 29(13):1602–1607, 2010.
- [Phillips *et al.*, 2017] Tyson Govan Phillips, Nicky Guenther, and Peter Ross McAree. When the dust settles: The four behaviors of lidar in the presence of fine airborne particulates. *Journal of Field Robotics*, 34(5):985–1009, 2017.
- [Suger *et al.*, 2016] Benjamin Suger, Bastian Steder, and Wolfram Burgard. Terrain-adaptive obstacle detection. *IEEE International Conference on Intelligent Robots and Systems*, 2016.
- [Urmson *et al.*, 2008] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner,

M. N. Clark, John Dolan, Dave Duggins, Tugrul Galatali, Chris Geyer, Michele Gittleman, Sam Harbaugh, Martial Hebert, Thomas M. Howard, Sascha Kolski, Alonzo Kelly, Maxim Likhachev, Matt McNaughton, Nick Miller, Kevin Peterson, Brian Pilnick, Raj Rajkumar, Paul Rybski, Bryan Salesky, Young-Woo Seo, Sanjiv Singh, Jarrod Snider, Anthony Stentz, William Red Whittaker, Ziv Wolkowicki, and Jason Ziglar. Autonomous Driving in Urban Environments: Boss and the Urban Challenge. *Journal of Field Robotics*, 25(8):524–466, 2008.

[Vidal-Calleja and Agammenoni, 2012] Teresa A. Vidal-Calleja and Gabriel Agammenoni. Integrated Probabilistic Generative Model for Detecting Smoke on Visual Images. *IEEE International Conference on Robotics and Automation*, 2012.

[Wurm *et al.*, 2009] Kai M. Wurm, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. Improving robot navigation in structured outdoor environments by identifying vegetation from laser data. *IEEE International Conference on Intelligent Robots and Systems*, 2009.

[Zhou and Tuzel, 2017] Yin Zhou and Oncel Tuzel. Vox- elnet: End-to-end learning for point cloud based 3d object detection. *CoRR*, 2017.