SIMULATION-BASED DATA AUGMENTATION USING PHYSICAL PRIORS FOR NOISE FILTERING DEEP NEURAL NETWORK

M. Jameela[a] , L. Chen[b], A. Sit[b], J. Yoo[a], C.Verheggen[b], G. Sohn[a*]

[a]Department of Earth and Space Science and Engineering, Lassonde School of Engineering
York University, Canada - (maryumja, jacobyoo, gsohn)@yorku.ca
[b]Teledyne Optech, Canada - (leihan.chen, andrew.sit, chris.verheggen)@teledyne.com

Commission TC II/3

KEY WORDS: Mobile Mapping System, Airborne LiDAR, Remote Sensing, 3D Representation, Noise Filtering, Systematic Noise, Data Augmentation

ABSTRACT:

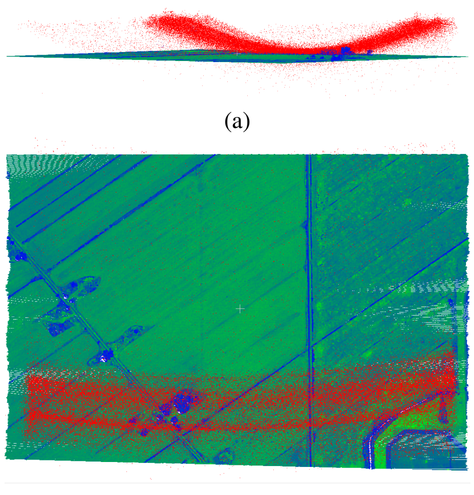
LiDAR (Light Detection and Ranging) mounted with static and mobile vehicles has been rapidly adopted as a primary sensor for mapping natural and built environments for a range of civil and military applications. Recently, technology advancement in electro-optical engineering enables acquiring laser returns at high pulse repetition frequency (PRF) from 100Hz to 2MHz for airborne LiDAR, which leads to an increase in the density of 3D point cloud significantly. Traditional systems with lower PRF had a single pulse-in-air zone (PIA) big enough to avoid a mismatch between pulse pair at the receiver. Modern multiple pulses-in-air (MPIA) technology ensures multiple windows of operational ranges for single flight line and no blind-zones; downside of the technology is projection of atmospheric returns closer to same PIA zone of neighbouring ground points and more likely to be overlapping with objects of interest. These characteristics of noise compromise the quality of the scene and encourage usage of noise filtering neural network as existing filters are not effective. A noise filtering deep neural network requires a considerable volume of the diverse annotated dataset, which is expensive. We developed simulation for data augmentation based on physical priors and Gaussian generative function. Our study compares deep learning networks for noise filtering and shows performance gain on 3D U-Net. Then, we evaluate 3D U-Net for simulation-based data augmentation, which shows an increase in precision and F1-score. We also provide an analysis of the underline spatial distribution of points and their impact on data augmentation, and noise filtering.

1. INTRODUCTION

LiDARs have emerged as powerful mapping tools for urban planning, navigation systems, and robotics. Airborne LiDAR has revolutionized the process of data acquisition for topographical surveying to capture reality. Sometimes, the perceived quality of the scene is compromised by unwanted reality. This unwanted atmospheric data, due to sensor operation or adverse weather conditions such as fog, rain, or snow, is broadly termed as noise.
Traditional systems with lower PRF can complete a survey in a single pulse-in-air zone, which avoided the mismatch of pulse pair at the receiver. As the technology advanced, gateless LiDAR sensors were introduced with higher PRF, no restriction of single window operational ranges, and no blind-zones guaranteeing high-density 3D point cloud. As PRF increases, PIA zones become narrower, which requires manufacturers to come up with algorithms to track the PIA zone automatically within a single flight-line to match laser pulse pair. The downside of the technology was atmospheric points are most likely to be projected closer to or overlap with objects of interest. Figure 1 shows the 3D point cloud acquired using Teledyne Optech's Galaxy T1000 airborne LiDAR to visualize the compromised perception of the scene due to noise. So, while the atmospheric returns always existed, at lower PRF, we could remove these atmospheric points with simple height filters, or nearest-neighbor algorithms quite effectively. But once they started mixing with the object of interest, more sophisticated algorithms become necessary.
Previous works have demonstrated encouraging performance to denoise 2D images using statistical, machine learning, and deep learning methods (Goyal et al., 2020). Existing sophisticated deep learning methods such as PointCleanNet (Rakotosaona et al., 2020) have dealt with sparse noise points for meshes. There are two significant studies for noise filtering due to adverse weather conditions for autonomous driving systems, which do not deal with sensor noise (Heinzler et al., 2019) and (Stanislas et al., 2019). Developing a deep neural network for noise filtering requires a thorough investigation of the diverse annotated dataset. We not only studied airborne LiDAR technology and its operation to understand the sensor noise but also acknowledge the need for a massive annotated dataset for training a deep neural network. The collection of the new dataset and manual annotation is labor-intensive and expensive. We developed simulations to replicate the noise mechanism to analyze the dataset and augment it for training noise filtering neural network. These simulations used physical priors and Gaussian generative function for producing synthetic noise with variable density. It considers the multiple pulses in air (MPIA) technology to estimate the distance R_{OBS} traveled by pulse beyond the maximum range before the next beam was fired (Roth and Thompson, 2008). R_{OBS} helps in determining the proximity of each point from their respective PIA zone. This proximity can be an indication of a point belonging to the systematic noise pattern. Significant contributions of our work are as follows:

- Case study of the sensor noise and simulation-based data augmentation; and
- Our work also demonstrates the performance of various deep neural networks for noise filtering and the impact of simulation-based data augmentation on deep neural network 3D U-Net (Çiçek et al., 2016).

*Corresponding author


(a)

(b)

Figure 1: 3D point cloud from Galaxy T1000 (a) Sideview, (b) Top view; red points represent noise

2. RELATED WORK

In this section, we will discuss the recent work related to 3D data augmentation. Data augmentation increases the volume and diversity of the dataset. Some of the data augmentation methods are relatively simpler such as scaling, colouring, shifting, rotating, while others are much more complex such as simulations and deep learning. We will divide this section into two significant subsections, simulation-based, and deep learning-based data augmentation (Shorten and Khoshgoftaar, 2019).

2.1 Deep Learning Networks

There are two approaches of deep learning often used for data augmentation; generative adversarial network (GAN) and adversarial training networks. PC-GAN is a novel approach to generate synthetic 3D object point cloud (Li et al., 2018). PC-GAN proposed a generic framework to use the underline distribution of data to create a point cloud for 3D models. Due to its usage of local latent variables to understand the neighborhood and spatial location of the point and global latent variable to interpret the overall shape of the object, it's not suitable for large scale point cloud generation. (Achlioptas et al., 2017) has proposed an autoencoder which produces a latent space that effectively increased the performance accuracy of GAN networks for 3D point cloud objects. PointFlow learns the distribution of shape and points and uses the invertible parametrized transformation to learn from these distributions and generate a model for syntenic data (Yang et al., 2019). This approach increases the accuracy of generating a 3D object point cloud. Adversarial examples are objects that look like real objects with few perturbations. These examples are generated using deep learning or simple geometrical manipulation to create synthetic datasets. GAN networks proposed to generate the synthetic 3D object point cloud are relevant for object detection and recognition(Shu et al., 2019). Existing frameworks lack the generation of synthetic data for large scale 3D point cloud. They also deal with convergence problems and suffer from difficulty in producing high-resolution output.

2.2 Simulations

There are different simulations developed to address the data scarcity for large scale point cloud using virtual reality and gaming. One such simulation generates synthetic scenes from the game-based environment (Yue et al., 2018). These scenes can be customized by the user, which can help boost the performance of neural networks training on a synthetic dataset. The results showed performance gain on a semantic segmentation task. (Sallab et al., 2019) proposed a hybrid technique that used real data to make simulated data more realistic using cycleGAN. It also shows that simulated data is not usually very realistic, especially by off-the-shelf opensource simulations for autonomous driving scenes. Commercial tool Blender's 3D sensor simulation plugin used to generate a large-scale 3D point cloud called SynthCity (Griffiths and Boehm, 2019a). There are multiple other data augmentation techniques based on class weights for imbalanced class distribution or random duplication of points proposed by (Griffiths and Boehm, 2019b) and (Qi et al., 2017). Research work for autonomous driving vehicles that deals with noise due to adverse weather conditions proposed augmentation model for fog and rain (Heinzler et al., 2019). It utilizes distance matrix, intensity matrix, extinction coefficient, and point scattering rate for the augmentation of rain and fog. The augmented dataset is used to train proposed neural network architecture and shows overall performance increase. Though it also caused ambiguity between rain and fog classes due to their inherently same nature. Our approach proposed a simulation to generate atmospheric points based on the principals of MPIA and increasing PRF to reflect the uniqueness of modern sensor noise.

3. METHODOLOGY

In this section, we will discuss our proposed synthetic noise simulations. To understand our proposed methodology, we first discuss the basic concepts of LiDAR scanning. We then introduced physical priors-based simulation and Gaussian model-based simulation for generating synthetic noise.

3.1 Airborne LiDAR Scanning:

LiDAR is built from these significant components; LiDAR sensor, a GPS receiver, and an inertial measurement unit (IMU) mounted on a vehicle (helicopter or plane) shows in Figure 2 (El-Sheimy, 2005). LiDAR scanner emits a laser pulse (echo/beam), which reflects from the target to the receiver. It calculates the 3D point using three major measurements; the position of the sensor, the direction in which the signal traveled, and the distance covered by the pulse for hitting the target. Trajectory information is acquired using a global navigation satellite system's receiver, which is mounted on the vehicle along with altitude and orientation. IMU is used to track the position of LiDAR using pitch, roll, and yaw angles. LiDAR's signal is deflected using a mirror inside the scanner, and the position of the mirror is stored on every laser pulse shot. (Rohrbach, 2015). The euclidean distance equation is used to calculate the distance R between the target and sensor.

$$R = \sqrt{(x_p - x_l)^2 + (y_p - y_l)^2 + (z_p - z_l)^2} \tag{1}$$

where x_p, y_p, z_p = coordinates of the points
x_l, y_l, z_l = coordinates of the sensor

One of the most critical parameters of the LiDAR is pulse rate frequency (PRF), which is the number of beams shot in one second. MPIA technology enables the sensor to fire the next beam before receiving the last. Manufacturers have the proprietary algorithm in place to match these pulse pairs by tracking the PIA zone automatically for a single flight. The negative side of the technology is that atmospheric points are projected closer to the

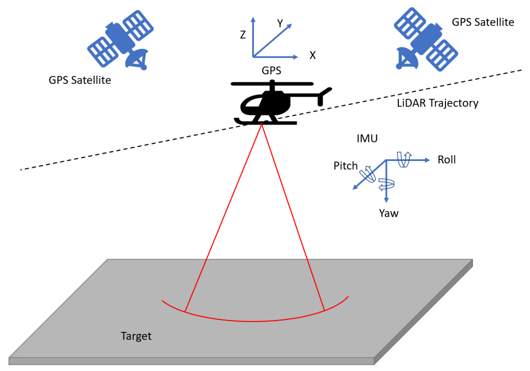


Figure 2: Airborne LiDAR

transition point of PIA zone of the neighboring regular/non-noise objects.The increasing PRF makes this problem worse as more shots, and the same rate of atmospheric returns per shot causes more atmospheric points. Higher PRF also develops narrower PIA zones due to which atmospheric points are likely to be closer to or overlaps with regular/non-noise objects. Figure 3 shows workflow of the simulations. Both simulations used PIA LiDAR equation for estimating physical priors. P2-Simulation uses travel time of laser pulse while GM-Simulation uses distance from sensor to target.

3.2 Physical Priors-based Simulation (P2-Simulation)

The simulation-based on physical priors assumes that travel time for atmospheric points is less than target data. PRF is taken from LiDAR configuration, and the maximum sensing range R_{MAX} is estimated by the speed of light c and PRF $\frac{c}{2\times PRF}$. The simulation model takes PRF, c, regular data points, and noise density and output simulated data. Figure 4 shows how simulation estimates the travel time of the atmospheric return. Opportunity time window $\frac{1}{PRF}$ gives time limit for receiving signal to correctly match with sent signal without specific matching algorithm for laser pulses. This time window helps in calculating travel time difference Δt between regular and synthetic noise points. Δt is used to estimate the travel time of synthetic noise and projects it on a transition point between two PIA zones.

3.3 Gaussian Model-based Simulation (GM-Simulation)

We selected Gaussian generative models for our second simulation to fix limitations of P2-Simulation by extracting range constraints, as shown in Figure 3. The simulation model takes PRF, c, regular data points, flight trajectory, and noise density to output simulated data based on Algorithm 1. Physical priors were calculated using PIA LiDAR equation; $PIA = ceil(\frac{R}{R_{MAX}})$ and R_{OBS} traveled by any pulse beyond maximum range R_{MAX}. The generative noise functions use these physical priors to estimate Gaussian parameters. The distribution of R_{OBS} from manually annotated data is used to generate random normal distribution of given noise density, which is used in calculating the distance R_{SN} between synthetic noise point and sensor using a PIA zone that of a particular regular/non-noise point. Synthetic noise is projected on the distance R_{SN} along the vector from the sensor to a regular/non-noise point, as shown in Figure 5.

where	N_{RI} = Total no. of raw input points
	N_{AN}= Total no. of actual noise points
	N_{SN} = Total no. of synthetic noise points
	N_{REG}= Total no. of points in clean point cloud

```
Algorithm 1: Gaussian Model-based Simulation.
Input: $PRF$, $c$, $N_{SN}$, $R^{[i]}_{REG}$,$T^{[i]}_{REG}$ $P^{[i]}_{REG}$, $R^{[j]}_{RI}$, $P^{[i]}_{REG}$,
  $P^{[j]}_{RI}$, $i = 1, 2, 3, ....., N_{REG}$,
  $j = 1, 2, 3, ......, N_{RI}$
Result: $P_{SD}$: GM-Simulated Data
$R_{MAX} = \frac{c}{2\times PRF}$;
$R_{AN} = R_{RI} - R_{REG}$;
while $k \leq N_{AN}$ do
    $PIA^{[k]}_{AN} = ceil(\frac{R^{[k]}_{AN}}{R_{MAX}})$;
    $R^{[k]}_{OBS} = R^{[k]}_{AN} mod R_{MAX}$;
end
$\mu_{OBS} = mean(R_{OBS})$;
$\sigma_{OBS} = standard\_dev(R_{OBS})$;
$R^{SN}_{OBS} = Gaussian(\mu_{OBS}, \sigma_{OBS}, N_{SN})$;
while $m \leq N_{SN}$ do
    $R^{[m]}_{SN} = ((PIA^{[m]}_{REG} - 1) \times R_{MAX} + R^{[m]}_{OBS})$;
    $P^{[m]}_{SN} = \frac{\overrightarrow{T^{[m]}_{REG} P^{[m]}_{REG}}}{|\overrightarrow{T^{[m]}_{REG} P^{[m]}_{REG}}|} \times R^{[m]}_{SN}$ ;
end
$P_{SD} = P_{SN} + P_{REG}$
```

$$T^{[i]}_{REG} = \{i \in \mathbb{R}^3; 0 < i \leq N_{REG}\}$$
$$P^{[i]}_{REG} = \{i \in \mathbb{R}^3; 0 < i \leq N_{REG}\}$$
$$P^{[j]}_{RI} = \{j \in \mathbb{R}^3; 0 < j \leq N_{RI}\}$$

3.4 Noise Filtering Deep Neural Network

We designed an experimental study using a deep neural network 3D U-Net to observe the results of noise filtering for 3D point cloud. The selection of 3D U-Net was based on its larger receptive field, performance efficiency, and state-of-art performance for semantic segmentation over various medical imaging datasets. The architecture of 3D U-Net can be seen in the Figure 6. It shows multi-resolution features extraction and decoding it to full resolution using skip connections.

The results of experiments and simulations lead us to understand the LiDAR noise characteristics. We also compared the performance of 3D U-Net with support vector machine (SVM), denoising autoencoder (DAE) (Palla et al., 2017), and PointNet (Qi et al., 2017). 3D U-Net is a deep network that requires massive, diverse datasets. We generated a simulation-based augmented dataset and trained 3D U-Net with and without augmentation to analyze performance differences. We only selected synthetic data from second simulation as it gives us fair opportunity of comparison due to its output similarity to real dataset as shown in Figure 3.

4. EXPERIMENTS

4.1 Datasets

For simulation and noise filtering experiments, we acquired a dataset of thirteen scenes of a site using the Teledyne Galaxy T1000. Each scene contains roughly 5 million points and coverage area of approximately $1km^2$. These are all outdoor rural scenes containing forest and agricultural land. All the scenes are manually labeled into two classes noise and regular objects. Due

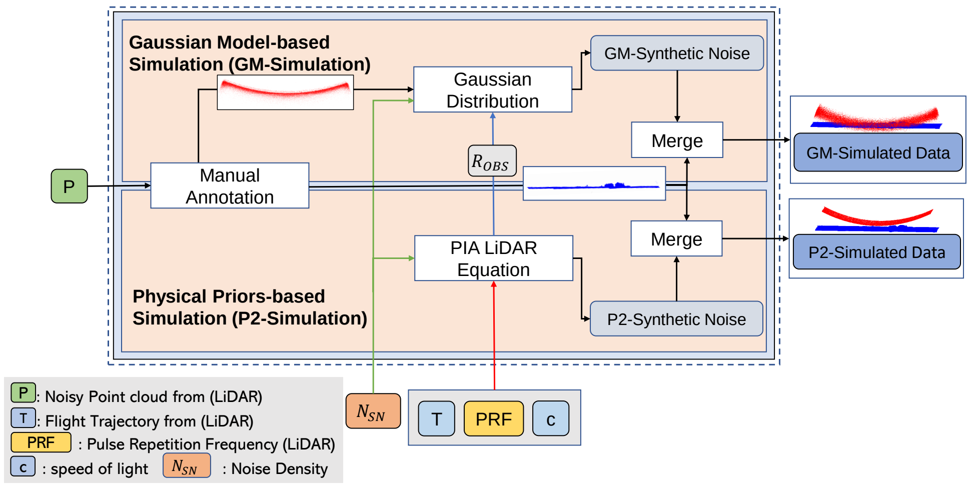

Gaussian Model-based Simulation (GM-Simulation)
Gaussian Distribution
GM-Synthetic Noise
Merge
GM-Simulated Data
P
Manual Annotation
R_{OBS}
PIA LiDAR Equation
Merge
P2-Simulated Data
Physical Priors-based Simulation (P2-Simulation)
P2-Synthetic Noise
P: Noisy Point cloud from (LiDAR)
T: Flight Trajectory from (LiDAR)
PRF : Pulse Repetition Frequency (LiDAR)
c : speed of light
N_{SN} : Noise Density
N_{SN}
T
PRF
c

Figure 3: Parametric Generative Model

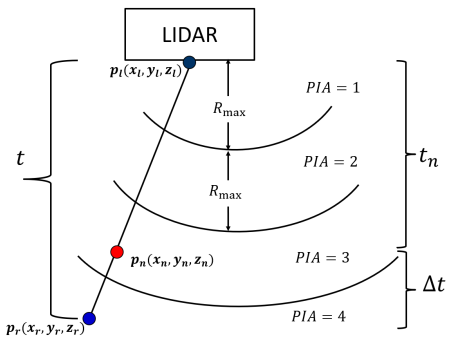


Figure 4: Physical Priors-based Simulation

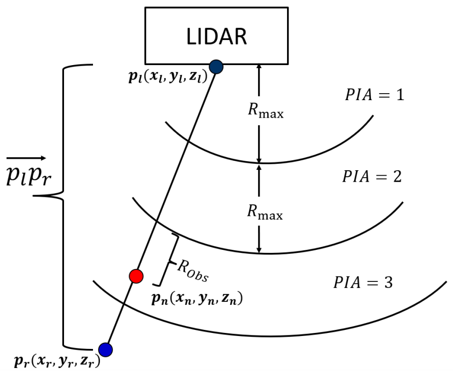


Figure 5: Gaussian Model-based Simulation

to 0.1% to 3.5% presence of noise, each scene has imbalanced class distribution.

4.2 Simulations

We took each manually labeled scene from the dataset; separate noise and regular objects from the scenes. Physical priors-based simulation takes the regular objects or clean LiDAR file to generate the synthetic noise along the transition point of PIA-zone. Gaussian model-based simulation requires real noise data to estimate the mean μ and standard deviation σ to generate the Gaus-

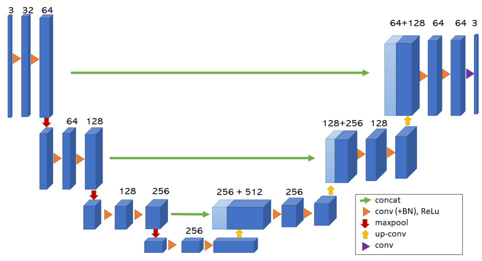


Figure 6: 3D U-Net Architecture

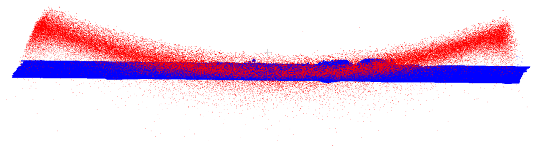
sian distribution of R_{OBS} for synthetic noise. Simulation package enables the user to either input a noisy ground truth for estimation of these parameters or manually input based on domain expertise.

4.3 Noise Filtering Using Simulation-based Data Augmentation

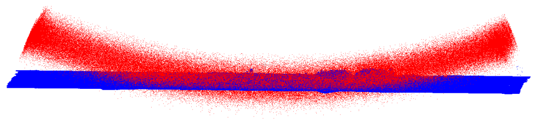
We randomly split the dataset into nine scenes for training and four scenes for testing. Each scene was then projected on the voxel grid, where each voxel is of size $2m^3$ and contains a count of points as feature. These voxels are then inputted as the cell of 128x128x128 to the network for noise filtering. We also trained 3D U-Net with a synthetic and real dataset together, which increased our dataset to 18 scenes. Synthetic data was generated using the GM-Simulation of 5% density of noise throughout the dataset. Both experiments have similar training settings.

5. RESULT AND DISCUSSION

Our experiments of sensor noise simulation and utilizing it for data augmentation for noise filtering has given interesting results. GM-Simulation simulated synthetic noise closer to real noise, as shown in Figure 7. To further verify our results, we calculated probability density function for noise shown in Figure 8, which validate similar spatial distribution of noise over R_{OBS}. We also performed experiments for noise filtering. Our first experiment with 3D U-Net shows excellent performance as compared to the SVM, DAE, and PointNet. We have compared the results on recall, precision, and F1-score for all test scenes for noise class.



(a) 1.2% Real Noise.



(b) 5% Synthetic noise

Figure 7: GM-Simulation vs Real data; red: noise points and blue: regular points

These matrices defined in the following Equations 2, 3, and 4.

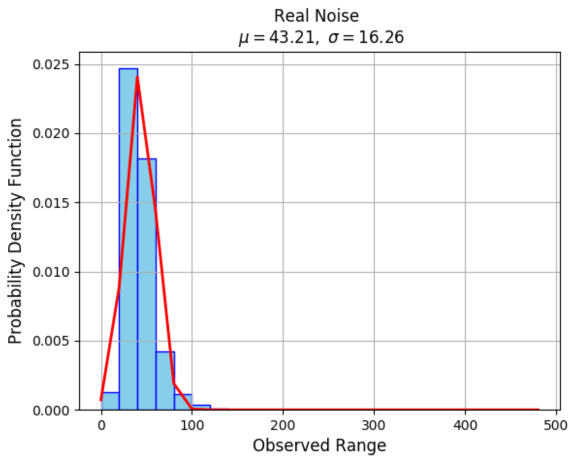
$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (2)$$

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (3)$$

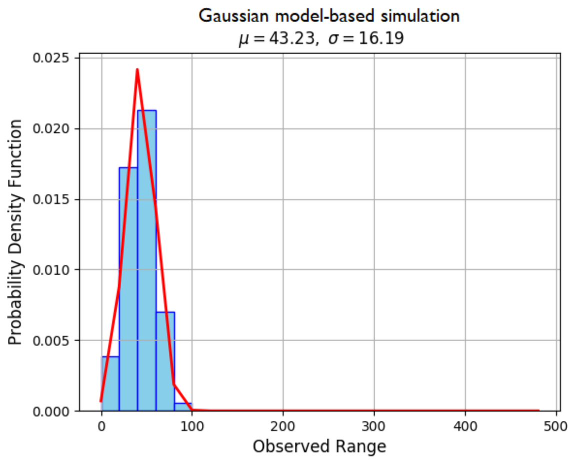
$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

Our ablation study of noise filtering shows that 3D U-Net has the best recall and F1-score for noise class and comparable precision, which are critical for noise filtering problems. Recall gives us correctly identified noise points from correctly predicted points. The F1-score provides a balance between completeness and correctness of the predicted noise points, as shown in Table 1. These results helps us in identifying the challenges of noise filtering as compare to semantic segmentation. Main reason of the failure of other methods is their lack of larger receptive field which is critical for context learning in regard to sensor noise containing global systematic noise pattern.
We then performed another experiment of 3D U-Net to observe the effects of data augmentation on the learning of the network. We trained network with synthetic and real data and tested on four real test scenes, as shown in Figure 14. We did not train our network with synthetic data by P2-Simulation. It generated noise on transition point between two PIA zones which did not show complex and random overlaps with objects of interest and would not be fair comparison. Empirically, a network that was trained using real and synthetic data outperformed vanilla training on precision and F1-score for almost all four cases. It has comparable performance on recall except for test scene 01, as shown in Figure 14 first row. Test scene 01 has denser noise as compared to other scenes, and training might have suffered from the bias due to similar noise density throughout the synthetic dataset. On the other hand, test scene 04 shows better performance on all three metrics for augmentation experiment , as shown in Table 2. It can be concluded from visualization of the results that augmentation helps high overlapping cases, as shown in row third and fourth of Figure 14. The results clearly show the problematic areas. Most of these noise and regular objects overlaps with each other or noise points clumped together, exhibiting the characteristics of complex objects such as trees or bushes. These results motivated us to observe the underlying global and local spatial distribution of points and understand the underline issues behind these results.

We calculated global spatial distribution of noise for all scenes which confirmed our assumption that noise is similarly distributed throughout the dataset. To clearly understand the



(a)



(b)

Figure 8: Comparison of the distribution of a) real noise from Figure 7-a) and b) GM-Simulation synthetic noise from Figure 7-b)

problem, we decided to investigate the local spatial distribution of noise points. We generated $20m^3$ voxel ten times the input voxel due to limitation of storage memory resources. Figure 9 reflects two different cases of noise and regular objects overlapping in a voxel. Graphs in Figure 10 indicate the probability density of noise points at R_{OBS} for the voxels from Figure 9-a) and 9-b). It is evident that the highest probability of random point to be noise is lower for Figure 9-b) then Figure 9-a) due to large overlaps and position of centroid.Graphs in Figure 11 show that probability for regular point to be at observed range interval of [34-47]m is 0 for Figure 9-a) but for Figure 9-b) its lowest for 34m and its highest for an interval of [42-46]m around transition point of PIA zone that is because of points distribution in the voxel is very close to noise. We observed two more voxels that only contain noise but with different density. The normal distribution of points in Figure 13 shows that the overall probability is higher for a Figure 12-b) voxel because of the density of points but lower for Figure 12-a). We concluded that noise present in the dataset shows various characteristics such as complexity, randomness and global systematic pattern and it can be divided into three major types; type-I: sparse noise, type II: systematic noise and type III: complex noise. It can help us compare the neural network generalization and performance with respect to noise types in a later stage.

	Noise		
	Recall	Precision	F-1 Score
SVM	46.78%	54.05%	0.501
DAE (Palla et al., 2017)	53.57%	**99.90%**	0.697
PointNet(Qi et al., 2017)	68.29%	98.19%	0.805
3D U-Net (Çiçek et al., 2016)	***77.92%***	*91.43%*	***0.841***

Table 1: Comparison of 3D U-Net for noise filtering with SVM, Denoising Autoencoder, and PointNet over recall, precision and F1-Score.

	Noise (w/o aug.)			Noise (aug.)		
	Recall	Precision	F-1 Score	Recall	Precision	F-1 Score
Test Scene 01	**93.80%**	**96.50%**	**0.951**	79.53%	**96.50%**	0.872
Test Scene 02	**93.70%**	90.70%	0.921	93.10%	**95.10%**	**0.941**
Test Scene 03	**43.20%**	17.6%	0.25	38.80%	**44.10%**	**0.408**
Test Scene 04	47.10%	23.50%	0.341	**53.80%**	**52.20%**	**0.530**

Table 2: 3D U-Net results for noise class using with and without augmentation for individual test scenes.

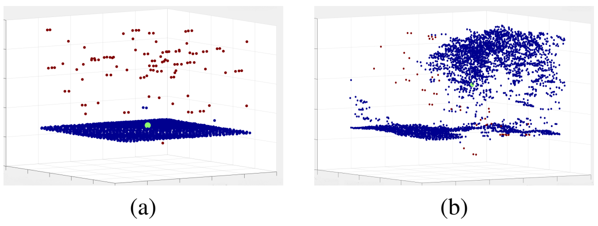


Figure 9: Two voxels from one scene, a) sparse noise far from the terrain, b) complex noise overlapping with bushes; red: noise points, blue: regular points and green: centroid

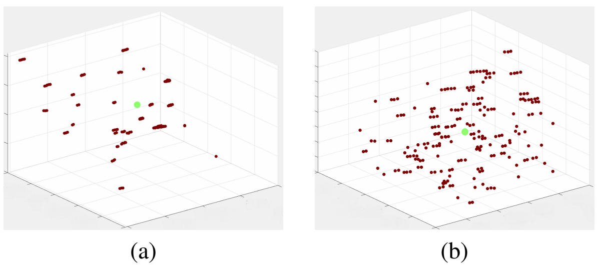


Figure 12: Raw point cloud of $20m^3$ relatively far from terrain containing only noise, a) sparse noise, b) dense noise relatively close to the terrain than a); red: noise points and green: centroid

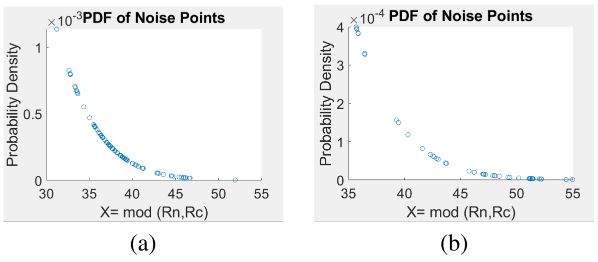


Figure 10: Normal distribution of noise points for voxels from Figure 9

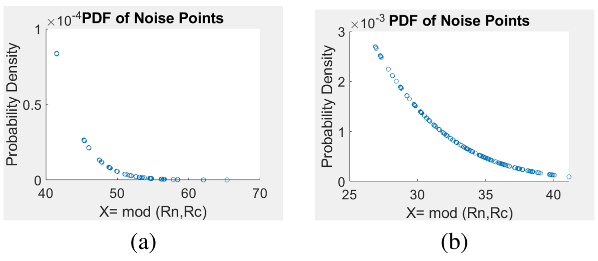


Figure 13: Normal distribution of noise points for voxels from Figure 12

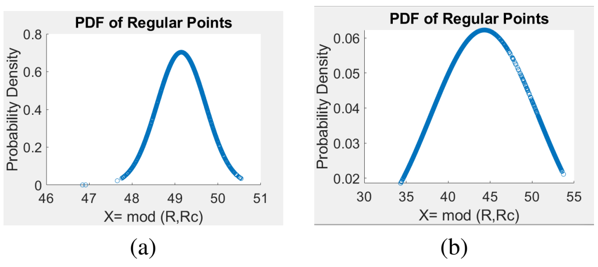


Figure 11: Normal Distribution of regular points for voxels from Figure 9

6. CONCLUSION

In this work, we reviewed the technology advancement of airborne LiDAR systems and their utilization in data acquisition for scientific and commercial applications. We concluded that physical priors, along with spatial distribution of points, provide leverage in simulating synthetic noise. We showed simulation-based data augmentation can improve performance for certain cases. We also evaluated the underline global and local distribution of noise points to better understand the results obtained for noise filtering. Our analyses state the importance of differentiating the noise in types; type I: sparse noise, type II: systematic noise, and type III: complex noise. In future works, we can utilize these analyses for designing a new deep neural network for noise filtering using a physical priors-based attention module.

ACKNOWLEDGEMENTS

This project, entitled “3D Mobile Mapping Artificial Intelligence (3DMMAI)”, is supported by the NSERC (Natural Sciences and Engineering Research Council of Canada) Collaborative Research Development (CRD) program and Teledyne Optech in Toronto.

REFERENCES

Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L. J., 2017. Representation Learning and Adversarial Generation of 3D Point Clouds. abs/1707.02392. http://arxiv.org/abs/1707.02392.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., Ronneberger, O., 2016. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI,2016)*, 9901, 424–432.

El-Sheimy, N., 2005. An Overview of Mobile Mapping Systems. *International Conference for Global Spatial Data Infrastructure (GSDI-8)*, 1–14.

Goyal, B., Dogra, A., Agrawal, S., Sohi, B. S., Sharma, A., 2020. Image denoising review: From classical to state-of-the-art approaches. *Information Fusion*, 55, 220–244.

Griffiths, D., Boehm, J., 2019a. SynthCity: A large scale synthetic point cloud. *Computing Research Repository*, abs/1907.04758. http://arxiv.org/abs/1907.04758.

Griffiths, D., Boehm, J., 2019b. Weighted Point Cloud Augmentation for Neural Network Training Data Class-Imbalance. *Computing Research Repository*, abs/1904.04094. http://arxiv.org/abs/1904.04094.

Heinzler, R., Piewak, F., Schindler, P., Stork, W., 2019. CNN-based Lidar Point Cloud De-Noising in Adverse Weather. *Computing Research Repository*. http://dx.doi.org/10.1109/LRA.2020.2972865.

Li, C., Zaheer, M., Zhang, Y., Póczos, B., Salakhutdinov, R., 2018. Point Cloud GAN. *Computing Research Repository*, abs/1810.05795. http://arxiv.org/abs/1810.05795.

Palla, A., Moloney, D., Fanucci, L., 2017. Fully convolutional denoising autoencoder for 3D scene reconstruction from a single depth image. *International Conference on Systems and Informatics (ICSAI,2017)*, 566–575.

Qi, C. R., Su, H., Mo, K., Guibas, L. J., 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *International Conference on Computer Vision and Pattern Recognition (CVPR,2017)*.

Rakotosaona, M. J., La Barbera, V., Guerrero, P., Mitra, N. J., Ovsjanikov, M., 2020. PointCleanNet: Learning to Denoise and Remove Outliers from Dense Point Clouds. *Computer Graphics Forum*, 39(1), 185–203.

Rohrbach, F., 2015. An Introduction to LiDAR. *english, personal blog*. https://felix.rohrba.ch/en/2015/an-introduction-to-lidar/.

Roth, R. B., Thompson, J. L., 2008. Practical Application of Multiple Pulse in Air (MPIA) Lidar in Large-area Surveys. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS)*, XXXVII(B1).

Sallab, A. E., Sobh, I., Zahran, M., Essam, N., 2019. LiDAR Sensor modeling and Data augmentation with GANs for Autonomous driving. *ICML Workshop on AI for Autonomous Driving*. http://arxiv.org/abs/1905.07290.

Shorten, C., Khoshgoftaar, T. M., 2019. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(60), 1-48.

Shu, D. W., Park, S. W., Kwon, J., 2019. 3D Point Cloud Generative Adversarial Network Based on Tree Structured Graph Convolutions. *International Conference on Computer Vision (ICCV, 2019)*.

Stanislas, L., Nubert, J., Dugas, D., Nitsch, J., Sünderhauf, N., Siegwart, R., Cadena, C., Peynot, T., Zurich, E., Zurich, S., 2019. Airborne Particle Classification in LiDAR Point Clouds Using Deep Learning. *International Conference on Field and Service Robotics (FSR, 2019)*, 1–14. https://eprints.qut.edu.au/133596/.

Yang, G., Huang, X., Hao, Z., Liu, M., Belongie, S. J., Hariharan, B., 2019. PointFlow: 3D Point Cloud Generation with Continuous Normalizing Flows. *International Conference on Computer Vision (ICCV, 2019)*.

Yue, X., Wu, B., Seshia, S. A., Keutzer, K., Sangiovanni-Vincentelli, A. L., 2018. A LiDAR Point Cloud Generator: from a Virtual World to Autonomous Driving. *International Conference on Multimedia Retrieval (ICMR, 2018)*, 458–464.

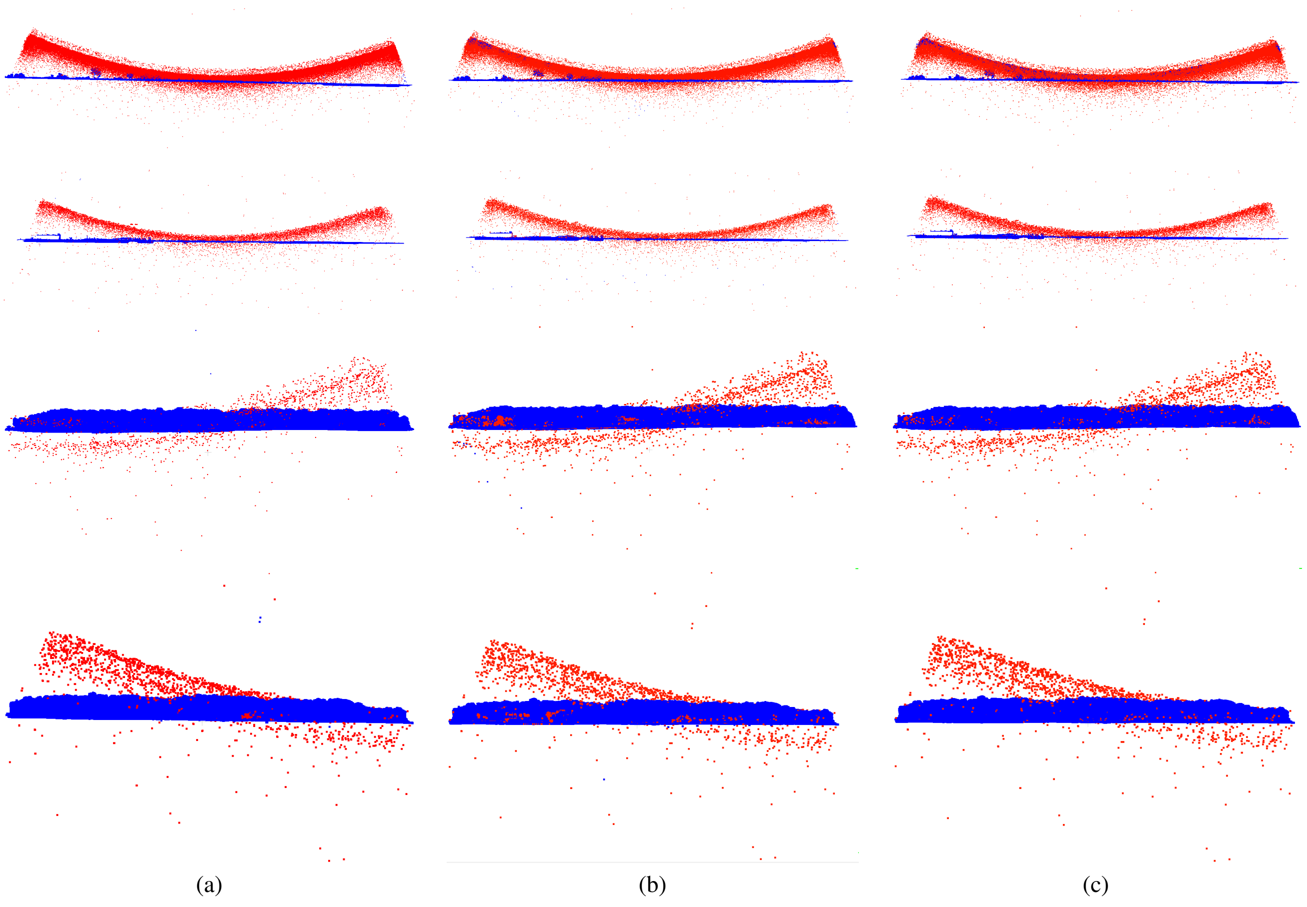


Figure 14: Visualization of results; a) Groundtruth, b) 3D U-Net (w/o aug.) and c) 3D U-Net (aug.). Red points are noise and blue points are regular non noise objects