

Article

# Application of Kalman Filter to Improve 3D LiDAR Signals of Autonomous Vehicles in Adverse Weather

Shih-Lin Lin \*  and Bing-Han Wu

Graduate Institute of Vehicle Engineering, National Changhua University of Education, No.1, Jin-De Rd., Changhua City, Changhua 500, Taiwan; M0734009@mail.ncue.edu.tw

\* Correspondence: lin040@cc.ncue.edu.tw

**Abstract:** A worldwide increase in the number of vehicles on the road has led to an increase in the frequency of serious traffic accidents, causing loss of life and property. Autonomous vehicles could be part of the solution, but their safe operation is dependent on the onboard LiDAR (light detection and ranging) systems used for the detection of the environment outside the vehicle. Unfortunately, problems with the application of LiDAR in autonomous vehicles remain, for example, the weakening of the echo detection capability in adverse weather conditions. The signal is also affected, even drowned out, by sensory noise outside the vehicles, and the problem can become so severe that the autonomous vehicle cannot move. Clearly, the accuracy of the stereo images sensed by the LiDAR must be improved. In this study, we developed a method to improve the acquisition of LiDAR data in adverse weather by using a combination of a Kalman filter and nearby point cloud denoising. The overall LiDAR framework was tested in experiments in a space 2 m in length and width and 0.6 m high. Normal weather and three kinds of adverse weather conditions (rain, thick smoke, and rain and thick smoke) were simulated. The results show that this system can be used to recover normal weather data from data measured by LiDAR even in adverse weather conditions. The results showed an effective improvement of 10% to 30% in the LiDAR stereo images. This method can be developed and widely applied in the future.

**Keywords:** autopilot; LiDAR of autonomous vehicles; Kalman filter

**Citation:** Lin, S.-L.; Wu, B.-H. Application of Kalman Filter to Improve 3D LiDAR Signals of Autonomous Vehicles in Adverse Weather. *Appl. Sci.* **2021**, *11*, 3018. <https://doi.org/10.3390/app11073018>

Academic Editor:  
Massimiliano Mattei

Received: 10 February 2021  
Accepted: 25 March 2021  
Published: 28 March 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Traffic accidents cause approximately 1.25 million deaths and hundreds of billions of dollars in economic loss worldwide each year. According to the World Health Organization, it is predicted that this will become the seventh leading cause of death in the world by 2030. The most important cause of traffic accidents is driver error. The limitations of human capabilities make it impossible for drivers to quickly make reasonable decisions in the face of emergencies. In addition, driver fatigue is also an important cause. With the development of artificial intelligence, along with improvements in computer and chip technology, it is possible to design autonomous vehicles, which could become one of the most important ways to reduce traffic accidents. Autonomous vehicles have stimulated a great deal of research interest [1–3].

Autonomous vehicles use LiDAR (light detection and ranging) systems to sense the environmental conditions outside the vehicle, but there are limitations to its use. The LiDAR imaging system can be affected by many factors, including the noise emitted and received by the LiDAR itself, noise caused by the movement of the vehicle, and background noise [4–7], which includes signals that are radiated or reflected to the receiving end by starlight, the earth, the sun, the atmosphere, clouds, and other sources of radiation. Adverse weather conditions, such as heavy rain, snow, and thick smoke, also produce background noise, which greatly reduce the detection ability of the LiDAR echoes. This study proposes combining a Kalman filter and neighboring point cloud segmentation algorithm to improve the acquisition of LiDAR data in inclement weather.

Kutilla et al. [8,9] proposed a plan for the development of LiDAR in 2016, comparing two different LiDARs (Ibeo Lux and Velodyne PUCK) in 2018. They evaluated the performance of the LiDAR in a fog chamber under stable foggy conditions. They found the performance to be affected, leading to a reduction in the visibility range under adverse weather conditions. Jiannan Chen et al. [10] proposed a 2D LiDAR algorithm for the detection and tracking of pedestrians. They studied pedestrian tracking and proposed a multi-person tracking algorithm based on the incorporation of a Kalman filter. Robin Heinzler et al. [11,12] proposed using a convolutional neural network (CNN)-based LiDAR system to filter out point cloud noise in bad weather. They observed that in heavy rain or thick smoke, water droplets may be misinterpreted as an object in front of the vehicle, causing the vehicle to stop. They focused on a 2D input layer method that usually uses a birds-eye view (BEV) or image projection view. The PointPillars introduced in this method are based on the use of a feature extraction network to generate a pseudo image from the point cloud, which is then used as input for the backbone of the CNN. This method outperformed the KITTI's dataset object detection in terms of detection performance and reasoning time. They proposed a 2D method inspired by LiLaNet's CNN architecture. Wallace et al. [13] introduced and discussed the incorporation of full-waveform pixel and image acquisition and processing that would allow the LiDAR to penetrate the obscuring medium and reconstruct a 3D map of the surface. Jokela et al. [14] tested and verified automotive point cloud sensors under severe weather conditions. They documented both indoor and outdoor conditions and tested how fog and snow would affect the performance of the LiDAR. They found the accuracy to also be reduced by exposure to environmental noise [15]. There are numerous examples of the impact of adverse weather conditions on LiDAR worldwide [16]. Consequently, there has been much research on how to improve LiDAR data [17,18].

Calibrated weather sensors have been used to measure the following relevant parameters: (1) transmission measurement of visibility in the range of 5 to 1000 m; (2) particle measurement of the particle size distribution of thick smoke from 0.4 to 40 microns; (3) rainfall intensity in the range of 0.001 to 1200 mm/h, as measured by a rain gauge and spectrometer. There have also been some studies on the efficacy of LiDARs of different wavelengths under different smoke densities. The same behavior is demonstrated at wavelengths from 350 to 1000 nm, regardless of the density or type of smoke, however, above 1000 nm, differences may appear. Thick smoke, especially convection fog, has an approximately 10% effect in the near-infrared range (1000–2400 nm). The wavelength is higher than the visible wavelength (400–800 nm) [19], so it is found that severe weather has some degree of influence on LiDAR [20].

In this current study, we first use LiDAR to make measurements under normal weather conditions and then simulate adverse weather conditions for rain, heavy smoke, and rain and heavy smoke. In this way, the difference in the LiDAR response for the different types of weather can be understood. An algorithm is developed to improve the signals received during adverse weather. The experiments show that the LiDAR results can be improved to be comparable to the signals received during normal weather. The results verify the effectiveness of the algorithm for making improvements in LiDAR systems for autonomous vehicles.

## 2. Research Methodology

The Kalman filter was first described in 1960 in a doctoral dissertation entitled *New Approach to Linear Filtering and Prediction Problems* published by Hungarian émigré Rudolf E. Kálmán [21]. Kalman filters are widely used for orbit calculation, target tracking, and navigation due to their real-time and anti-interference advantages as well as rapid action and high efficiency. In addition, they are also applied in the fields of integrated navigation and dynamic positioning, sensor data fusion, microeconomics, and so forth, especially for digital image processing and the currently hot research fields of pattern recognition, image segmentation, and image edge detection [22]. The Kalman filter is

basically a recursive algorithm, with each recursion containing two steps. First, an estimated value is calculated, and then a weighted summation of the estimated value and the measured value is obtained to get the best resultant state. The process comprises five steps with five corresponding formulas: (1) predicts the prior state  $x'_k$  using the corrected estimate  $\hat{x}_{k-1}$  and using the prior state  $\hat{x}_{k-1}$  predicted at time  $k-1$  to calculate the estimated value  $x'_k$  at time  $k$ ; (2) uses the noise  $P_{k-1}$  of the resulting state at time  $k-1$  to calculate the estimated noise  $P'_k$  at time  $k$ ; (3) uses the estimated value of the noise  $P'_k$  at time  $k$  and the measured value of noise  $r$  at time  $k$  to calculate the Kalman gain  $k_k$  at time  $k$ ;  $\hat{x}_k$  is the posterior estimate of the state  $k_k$  after correction of the prior estimate  $x'_k$  with the current measurement  $z_k$  obtained using (4);  $P_k$  is the error covariance of the estimated value  $\hat{x}_k$ , which is computed by correcting the previous estimated value  $P'_k$  with (5), where  $H$  is the parameter of the measurement system  $Q$  and  $R$  the process and the measurement noise covariance. The Kalman filter model assumes that the true state at time  $k$  is evolved from the state at time  $(k - 1)$ , which conforms to the following:

$$x'_k = Ax_{k-1} + Bu_{k-1} \tag{1}$$

$$P'_k = AP_{k-1}A^T + Q \tag{2}$$

$$k_k = P'_k H^T (HP'_k H^T + R)^{-1} \tag{3}$$

$$\hat{x}_k = x'_k + k_k(z_k - Hx'_k) \tag{4}$$

$$P_k = (1 - k_k H) P'_k. \tag{5}$$

Here,  $A$  and  $B$  are system parameters. Initially,  $x_0$  is a random vector, with an average value  $u_0 = E[x_0]$  and covariance  $P_0 = E[(x_0 - u_0)(x_0 - u_0)^T]$ . Here,  $w_k$  and  $v_k$  are the process noise and observation noise, both of which are assumed to be zero-mean multivariate Gaussian noise;  $u_k$  is the control vector. The function  $f$  can be used to estimate the previously predicted value, and the function  $h$  can be used to calculate the measured value from the predicted value. The covariances are, respectively,  $Q_k$  and  $R_k$  expressed as follows:  $E[w_k w_k^T] = Q_k$  and  $E[v_k v_k^T] = R_k$ .

The nearest neighbor segmentation algorithm is described below. This filtering method uses statistical filtering and radius filtering to remove noise. Statistical filtering refers to statistical analysis of the distance between the query point and the neighboring point cloud, and removes some noise that is not within the set range [23]. The  $n$ -th point in the LiDAR point cloud model is marked  $q_n, n = 1, 2, 3, \dots, S$ . Assuming that the distance from  $q_n$  to any point is  $d_i$ , then the average distance from  $q_n$  to all its  $k$  neighboring points  $D_{avr-n}$  is represented by the Gaussian distribution (the mean is  $\mu$ , the standard deviation is  $\sigma$ ) as in [24]:

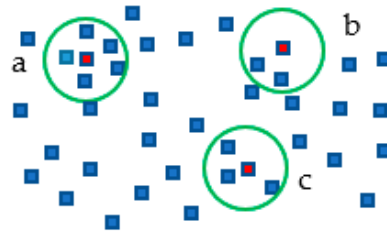
$$D_{avr-n} = \frac{\sum_{i=1}^k d_i}{k}. \tag{6}$$

Set the standard range  $S_{pa-n}$  to determine whether the point cloud in the model is noise. The calculation is as follows:

$$S_{pa-n} = \mu \pm g \cdot \sigma, g = 1, 2, \dots \tag{7}$$

Select the corresponding point according to the average distance  $D_{avr-n}$  and the standard range  $S_{pa-n}$ . That is, when the average distance  $q_n$  corresponding to the point  $D_{avr-n}$  is greater than the standard range  $S_{pa-n}$ , the point is deleted; otherwise, the point is retained [24]. Radius filtering is used to further eliminate noise. Radius filtering refers to the point cloud model stipulating that each data point must have at least a certain number of nearest neighbors within a specified range. Assuming that in the point cloud model, data point  $q_n$  is the center, then the number of nearest neighbors within a sphere whose radius is  $r$  is  $m$ , and the minimum number of nearest neighbors is set to  $M$ . According to  $m$  and  $M$ , judge whether to keep the corresponding point  $q_n$ . When  $m$  is greater than  $M$ , keep

the point; otherwise, remove the point [21]. The description is shown in Figure 1. When the radius (indicated by the green circle) specifies that there are at least three neighbors, only point b will be deleted. If it is specified that there must be at least five neighbors, both point a and point b will be deleted.



**Figure 1.** Principle sketch map of the radius filter.

### 3. Laboratory Equipment

The Velodyne LiDAR PUCK VLP-16 used in this study, as shown in Figure 2, allows full-time data transmission and reception, 360-degree full coverage, 3D distance measurement, and calibration reflection measurement. The farthest effective range of the VLP-16 sensor is 100 m, and the nearest effective range is 30 cm. It has low power consumption, is light weight, and has a small footprint and dual return functions. The structure of the LiDAR includes the LiDAR host, central processing system, power supply, and computer, as shown in Figure 3. Figure 4 illustrates the process of LiDAR signal transmission. An EFG-1400 wireless remote-control fog machine was used to simulate thick smoke; see Figure 4. This model has a remote-control distance of 8–10 m. The power is 1500 cu ft/min.



**Figure 2.** Velodyne LiDAR PUCK VLP-16.



**Figure 3.** Processing architecture of the LiDAR (light detection and ranging).



Figure 4. Smoke-making machine.

#### 4. Algorithm Simulation Verification

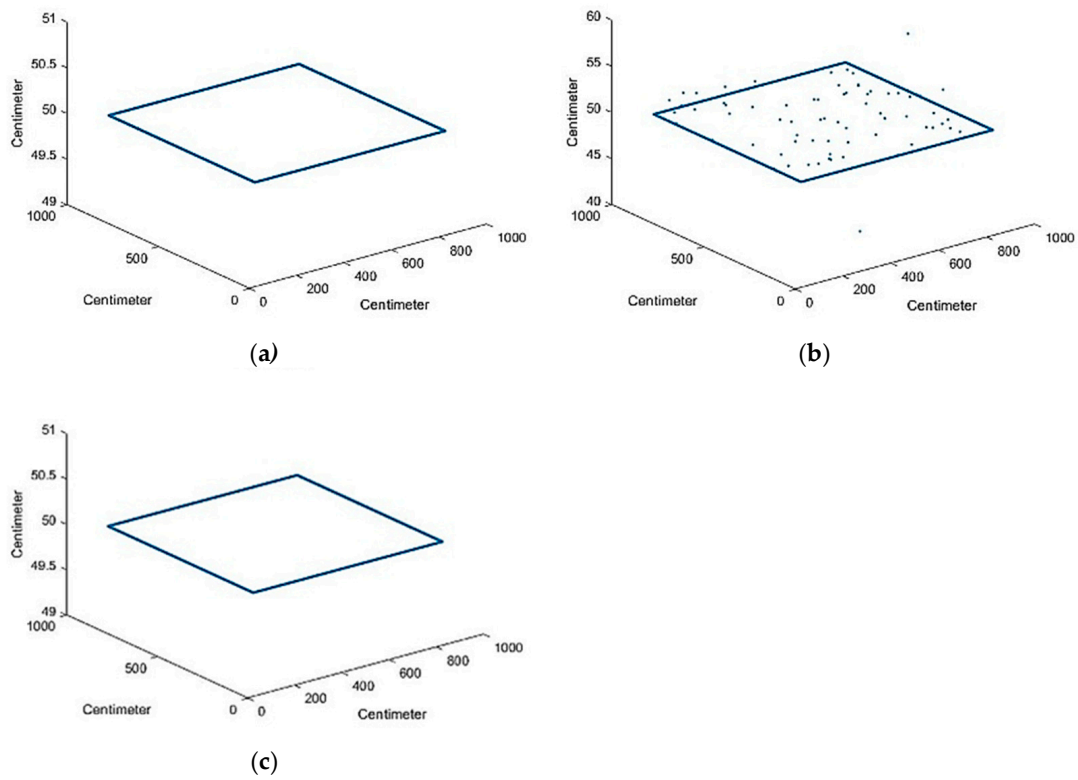
Two algorithms were used in this study: the neighbor point cloud segmentation algorithm and the Kalman filter. The MATLAB software was used to combine calculations for the two algorithms. The filtering algorithms were written for the following:

- (1) Simulation space;
- (2) Simulate the interference of rain or snow, which will cause noise in the space;
- (3) Neighboring point cloud segmentation algorithm;
- (4) Simulate normal weather LiDAR measurement;
- (5) LiDAR measurement disturbed by adverse weather;
- (6) Kalman filter reconstructs the LiDAR measurement results disturbed by adverse weather.

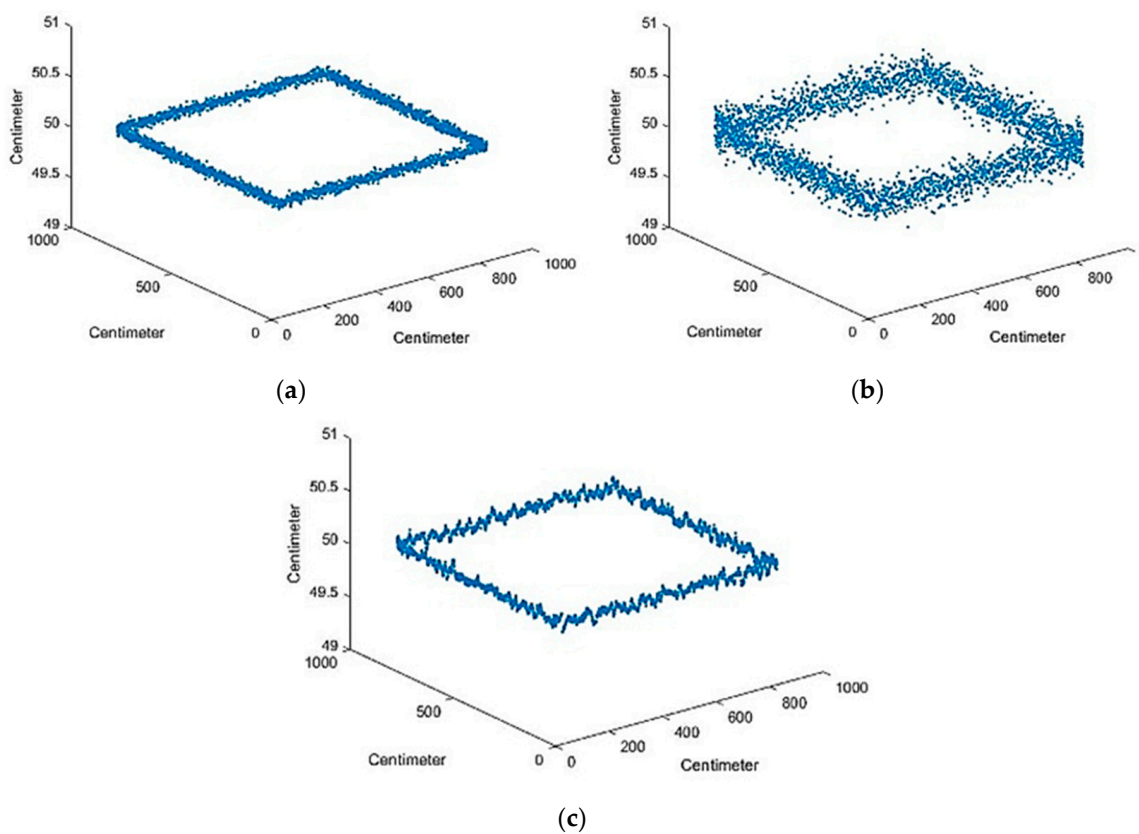
In the calculations, the size of the space was first established, which was set to 800 cm on the  $x$ -axis and the  $y$ -axis; the height of the  $z$ -axis is 50 cm, as shown in Figure 5a. Some point cloud noise was added to this space to test whether the two algorithms are appropriate. Nearest neighbor segmentation was tested first, then some point clouds were added for noise processing, as shown in Figure 5b. Using radius filter for filtering, we set the radius to be within 2 cm and the number of neighboring point clouds to be at least 10. Anything less than 10 would be filtered out as noise. The filtered result is shown in Figure 5c.

Figure 6a shows the simulated LiDAR signal. First, the LiDAR signal was simulated using the software. Gaussian white noise was simulated. The mean parameter was set to 0, and the parameter for the random number of standard deviations was set to 0.03. Real LiDAR will also produce a little error during measurement, but the value of the error will be within a certain range. Figure 6b shows the LiDAR signal with noise. In future, we will add Gaussian white noise for testing. The mean parameter was set to 0, and the random number of standard deviations was set to 0.1. Figure 6c shows the simulation results after Kalman filtering. The selection of  $Q$  and  $R$  is related to the statistical characteristics of the measurement and system noise. The current theoretical approach is to make an estimation based on experimental data. The parameters of the Kalman filter were set to matrix  $[1,0,0,0; 0,1,0,0]$ ,  $Q$  was set to a  $4 \times 4$  matrix, and  $R$  was set to  $[2,0; 0,2]$ . The five steps comprising Equations (1)–(5) of the Kalman filter process were calculated to restore the original data. The input was noise. We first calculated the estimated value and the estimated value of the noise, then the Kalman gain. The original data were restored after the Kalman gain was used to calculate the best result.





**Figure 5.** (a) Simulation space; (b) neighboring point cloud noise; (c) filter results of neighboring point cloud segmentation algorithm.



**Figure 6.** (a) LiDAR measurements; (b) LiDAR measurements disturbed by environmental noise; (c) Kalman filter reconstructed environmental noise.

## 5. Experimental Setting

In the experiments, a space 2 m long, 2 m wide, and 0.6 m high was constructed. The LiDAR was placed in the middle, 30 cm above the ground, as shown in Figure 7. The space was surrounded by water pipes to simulate the rain, and a fog-making machine was used to simulate the heavy smoke. The top of the space was sealed with plastic film, as shown in Figure 8. Three types of weather conditions were simulated: normal weather, heavy smoke, and rainy weather with heavy smoke. The response under the different weather patterns was studied, and the algorithms applied to improve the signal. Under normal conditions, the LiDAR signal was good and undisturbed, as shown in Figure 9. Figure 10 shows a flowchart of the experimental procedure.

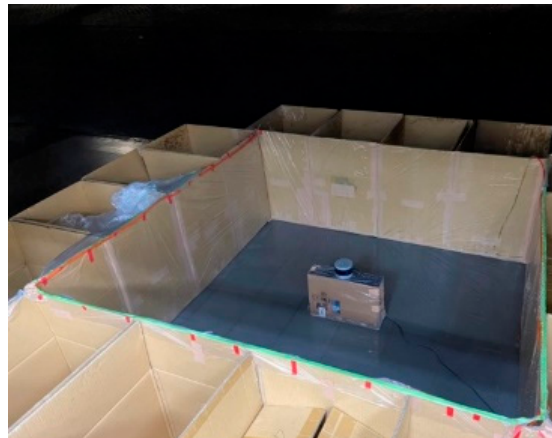


Figure 7. LiDAR measurement space.

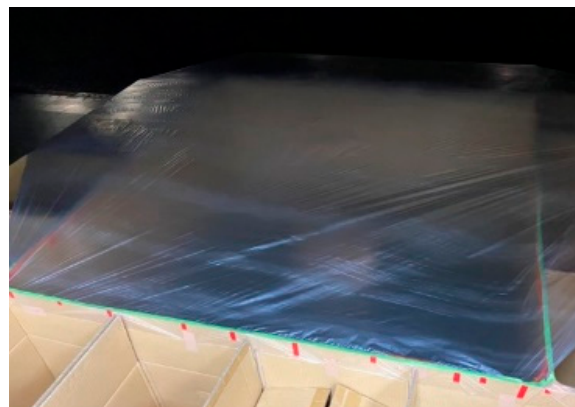


Figure 8. Simulated adverse weather site.

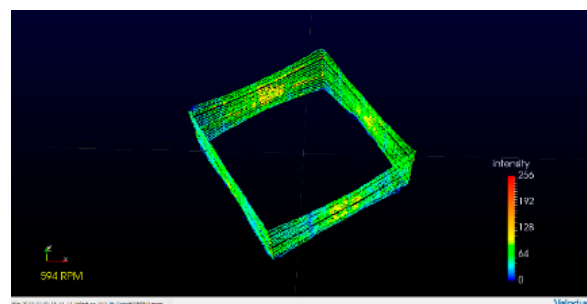
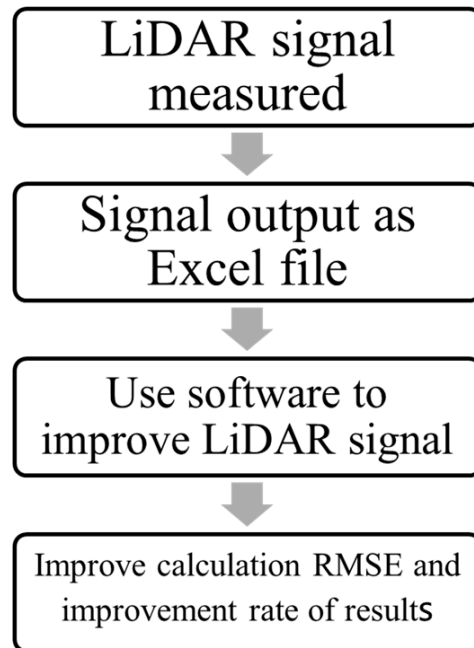


Figure 9. LiDAR in a normal state.



**Figure 10.** Experimental flowchart.

When simulating rainy weather, the size of the raindrops will also affect the LiDAR signal. If the raindrops are too small, the effect on the signal will not be obvious. Therefore, the simulated raindrops were made larger to achieve the required level of interference, as in Figure 11. In the second condition, dense smoke was simulated using the fog-making device, as in Figure 12. It was found that dense smoke has a considerable impact on LiDAR measurement. The denser the smoke, the more the light is dispersed to the surrounding environment. The third type of weather condition was both rain and heavy smoke, as in Figure 13. The measurements show that the LiDAR was more affected under the influence of two types of weather at the same time rather than when exposed to any single weather condition. The LiDAR's measurement capability was significantly reduced. Heavy smoke interfered so much with the signal that there was no signal in the four corners of the space.



**Figure 11.** Simulated rainy weather.





Figure 12. Simulated dense smoke.



Figure 13. Simulated rain and dense smoke.

The root mean square error (*RMSE*) was calculated to find the error value for the operation as follows:

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (\hat{y}_t - y_t)^2}{n}}, \quad (8)$$

where  $\hat{y}_t$  represents the data under normal weather conditions and  $y_t$  is the data under the influence of severe weather conditions, or the data after improvement. The larger the value of the RMES, the greater the deviation from the original data; the smaller the value obtained, the smaller the deviation from the original data. Comparison shows that the error values of the  $x$ -axis and the  $y$ -axis are very small. Only the error value of the  $z$ -axis is relatively large, so only the  $x$ -axis and the  $y$ -axis were compared.

## 6. Research Results

The above experimental test data (LiDAR point cloud data) were input into the software for calculation. The results under normal weather conditions are displayed in Figure 14. The results for rain are shown in Figure 15. The neighbor point cloud segmentation algorithm described in this research filtered out some noise, as shown in Figure 16. Finally, after passing through the Kalman filter, excess noise was filtered out, as shown in Figure 17. The resultant image obtained after using the algorithm is very similar to that obtained under normal weather conditions.

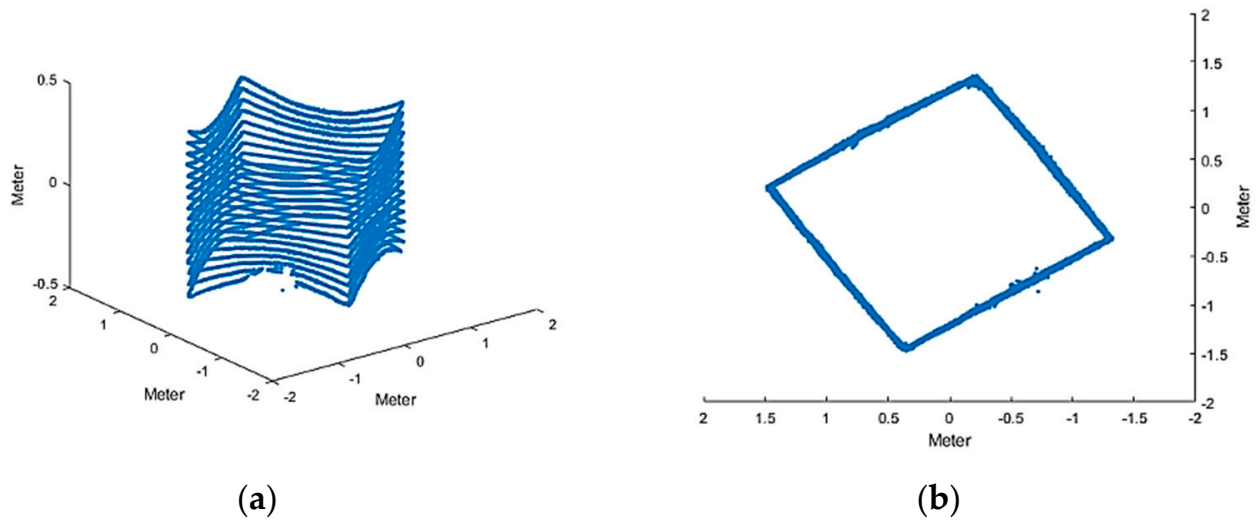


Figure 14. (a) 3D view and (b) top view under normal weather conditions.

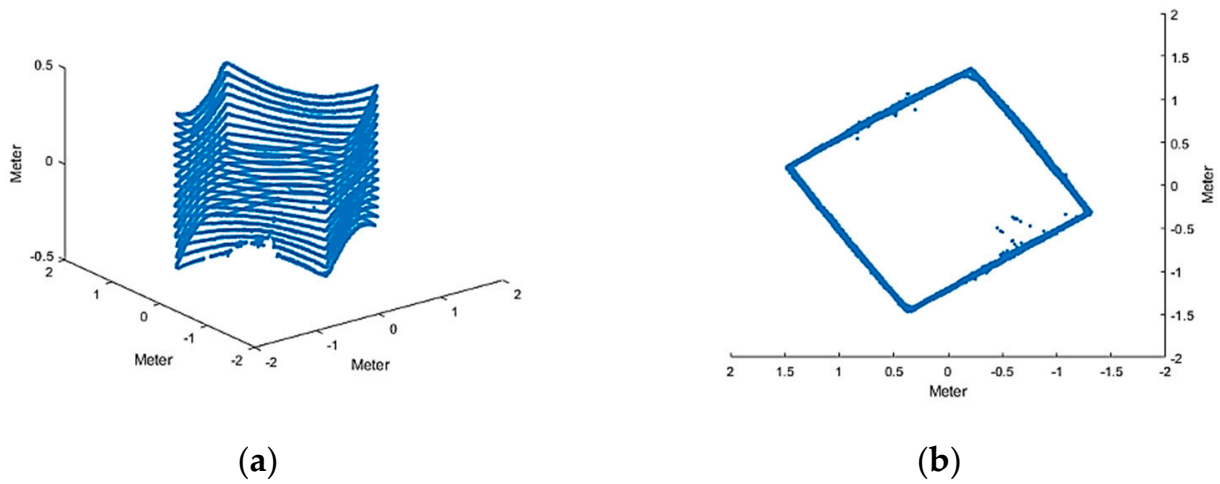


Figure 15. (a) 3D view and (b) top view for a simulated rainy day.

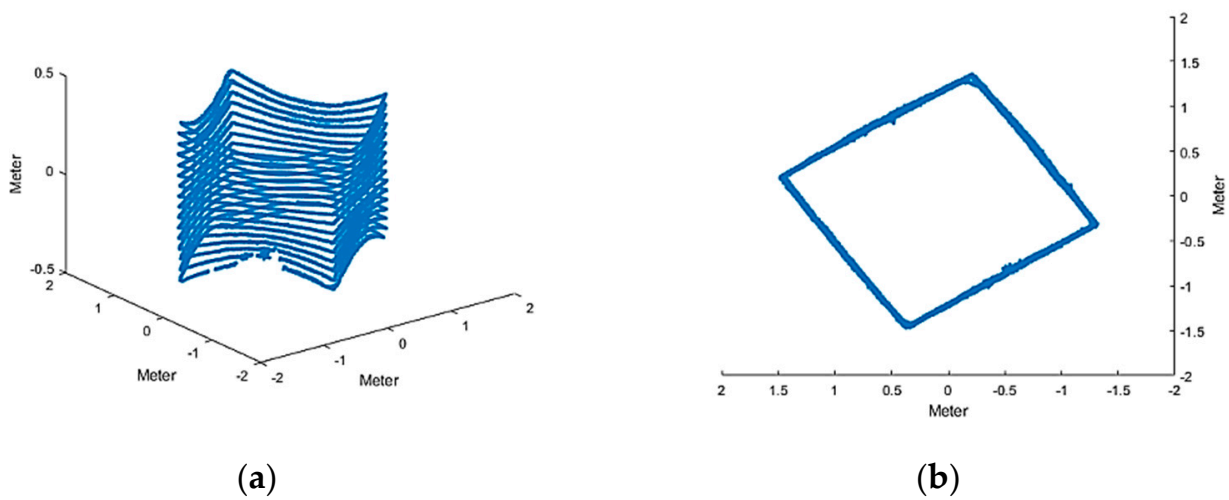
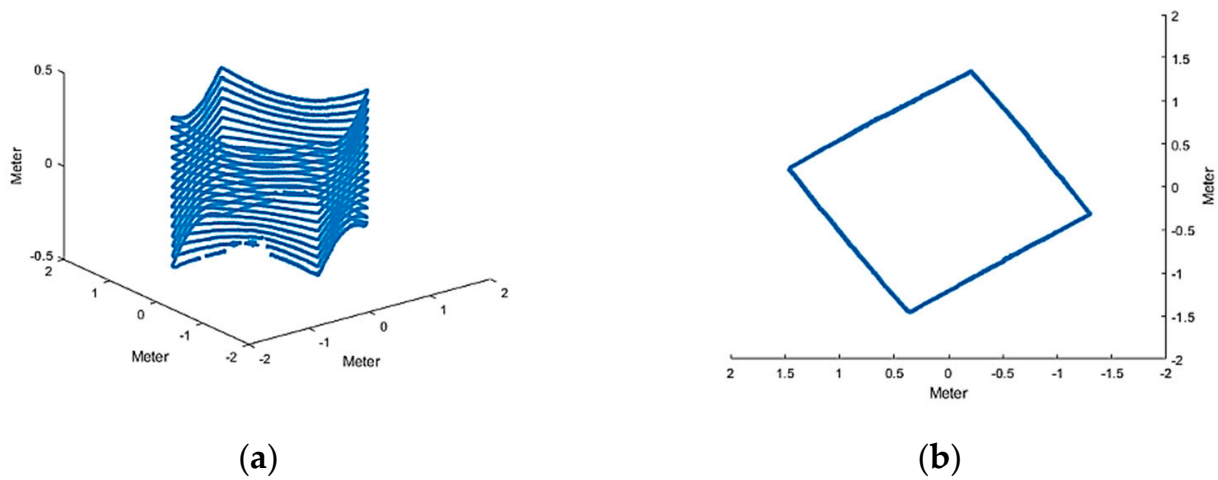


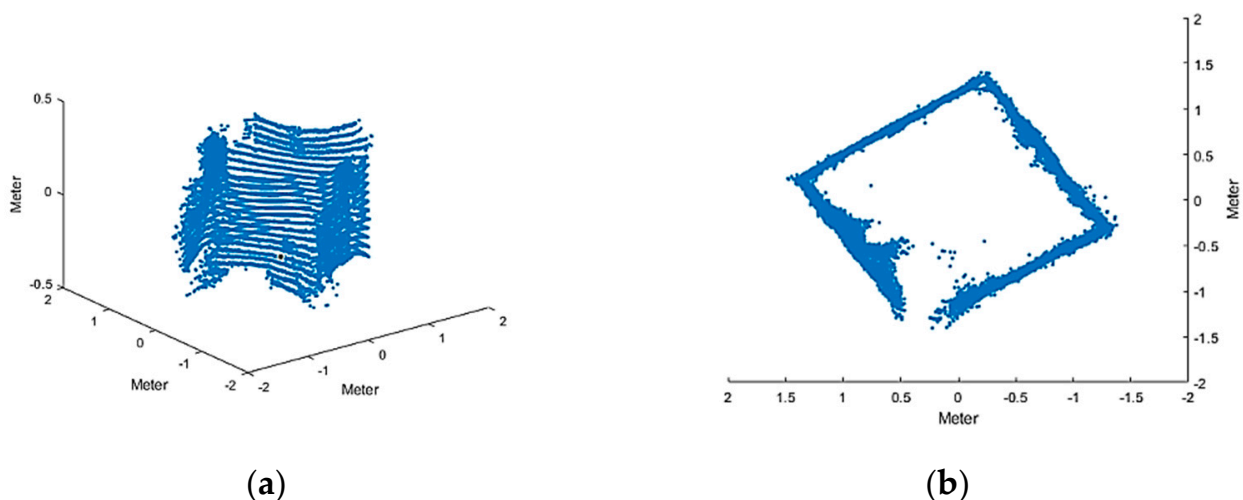
Figure 16. Filter results from the neighboring point cloud segmentation algorithm: (a) 3D view and (b) top view.



**Figure 17.** Kalman filter filtering results: (a) 3D view and (b) top view.

Next, the resultant improvement under heavy smoke conditions is explained. Figure 14 shows the 3D view and top view obtained in the normal situation. The influence of dense smoke can be seen on the image displayed in Figure 18. The lower corner is seriously affected by the interference of dense smoke. The improvement obtained by using the first nearest neighbor segmentation algorithm is shown in Figure 19. The result of improving the filtering model through the Kalman filter is shown in Figure 20. Although the process described in this study cannot restore the signal to that of normal weather, there was a significant improvement, and a lot of noise could be filtered out.

The third type of weather examined is the influence of both rain and heavy smoke, as shown in Figure 21. The results after processing with the nearest neighbor segmentation algorithm are shown in Figure 22. Although it was not possible to filter out all of the noise, it was possible to filter out most of it. Further improvement was obtained after Kalman filtering, as shown in Figure 23. The improvement is obvious, resulting in restoration of the four corners of the space. A lot of noise has been filtered out.



**Figure 18.** (a) 3D view and (b) top view under heavy smoke.

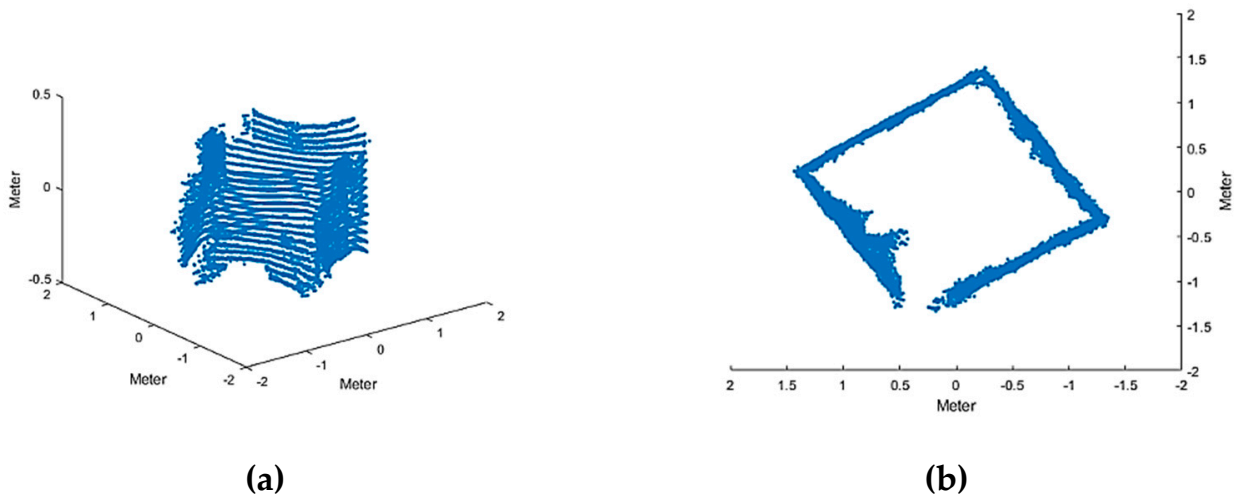


Figure 19. (a) 3D view and (b) top view of filtering results after neighboring point cloud segmentation.

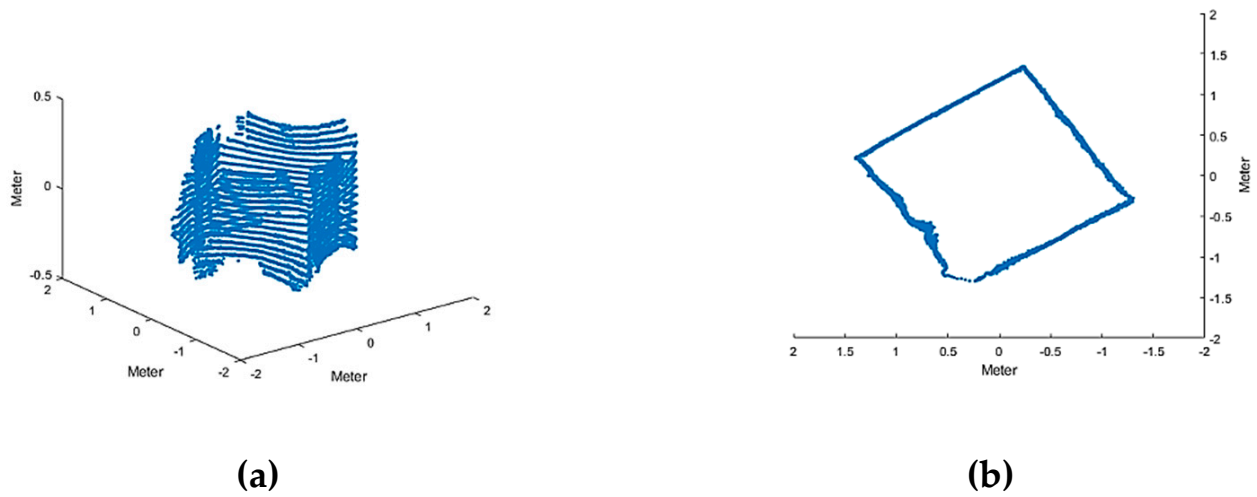


Figure 20. (a) 3D view and (b) top view of Kalman filter filtering results.

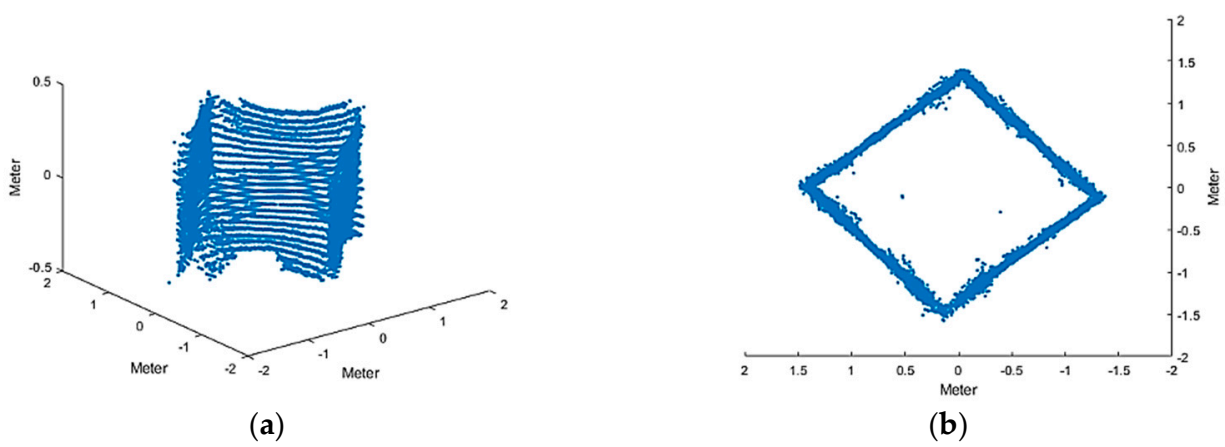
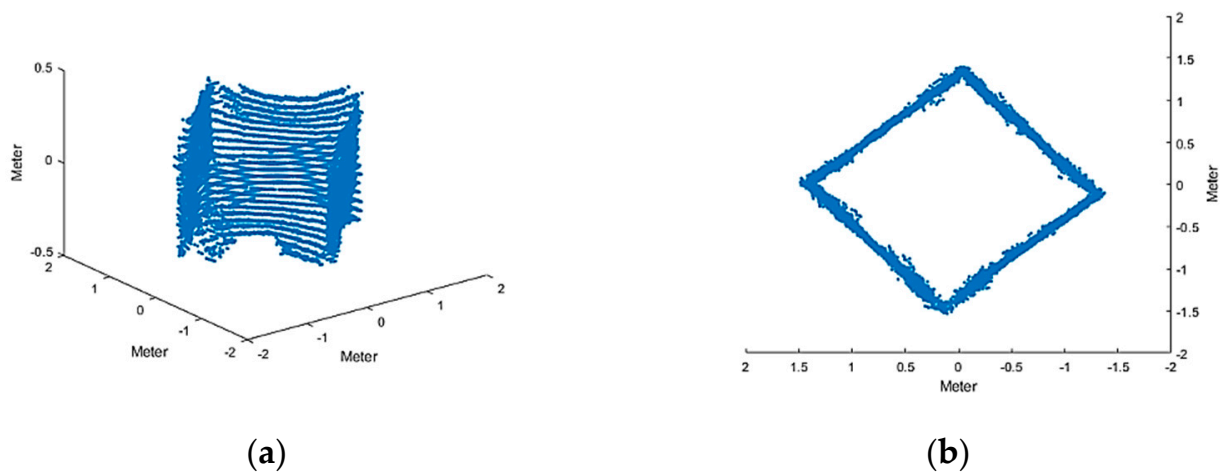
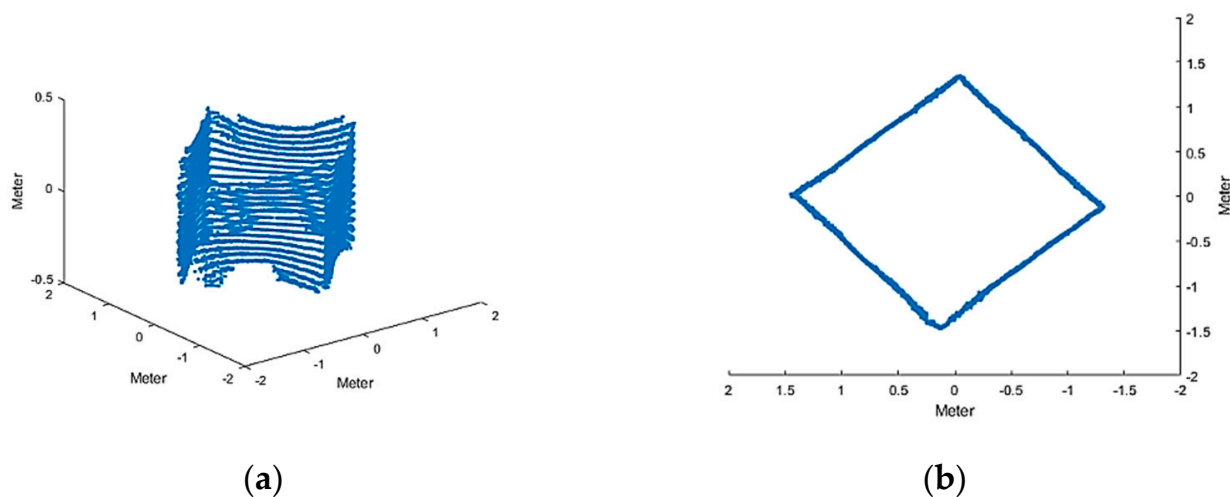


Figure 21. (a) 3D view and (b) top view under heavy smoke and rain.



**Figure 22.** (a) 3D view and (b) top view of filtering results after neighboring point cloud segmentation.



**Figure 23.** (a) 3D view and (b) top view after Kalman filtering.

The point cloud model data discussed in this research include an environmental frame with an  $x$ -axis,  $y$ -axis, and  $z$ -axis. The point cloud is located within the environmental frame. In the process of nearest neighbor point cloud segmentation within the environmental frame, a number of sampling circles were used to sample the point cloud. Two adjacent sampling circles overlap locally. When the number of point clouds obtained within the sampling circle was greater than a critical judgment value, the sampling circle was retained. The sampling circle has a radius of 2 cm. The critical judgment value is 10. When the number of point clouds in the sampling circle was less than the critical judgment value, the point cloud was filtered out as noise. In addition, Kalman filtering was used to improve the spatial model data to filter and output low-noise environment model data. As a result, the interference of environmental noise on LiDAR sensing was reduced. Kalman filtering was especially helpful to filter out the noise caused by dense smoke.

In order to understand the improvement demonstrated in the experiments, error analysis was conducted. For each weather condition and the improved results, the same number of point clouds were used for comparison, as shown in Table 1. The results show that the nearest neighbor segmentation algorithm and Kalman filter have lower RMES values.



**Table 1.** Comparison of RMSE between simulated normal weather and adverse weather before and after improvement (unit: meters).

Comparison of Normal Weather and Adverse Weather.	Adverse Weather and Used Algorithms	x-axis	y-axis
Rainy weather	Rain	0.0164	0.0134
	Nearest neighbor segmentation algorithm	0.0132	0.0103
Smoky weather	Kalman filter	0.0109	0.0087
	Smoky weather	0.0750	0.0849
	Nearest neighbor segmentation algorithm	0.0743	0.0841
Rainy weather and smoky weather	Kalman filter	0.0634	0.0646
	Rainy weather and smoky weather	0.0199	0.0313
	Nearest neighbor segmentation algorithm	0.0198	0.0312
	Kalman filter	0.0166	0.0277

From the above table, it can be seen that after using the nearest neighbor segmentation algorithm, there was significant improvement in the error value under normal conditions. After Kalman filtering, the error value was further reduced. After correcting the error value and comparing it with the normal value, there were obvious corrections. The improvement rate can be calculated as follows:

$$\text{Improvement rate} = \frac{(\text{Original weather} - \text{after improvement})}{\text{Original weather}} \cdot 100\% \quad (9)$$

where original weather represents the RMSE of the original weather, and after improvement represents the RMSE with improvement results. This equation was used to calculate the improvement rate under all three weather conditions. The two axes were corrected by about 30% for the rainy weather simulation, by about 15% in the case of heavy smoke, and by about 11% in the rainy weather and heavy smoke simulation, as shown in Table 2.

**Table 2.** Improvement rate after correction under three severe weather conditions.

	Rainy	Smoky Weather	Rainy Weather and Smoky Weather
x-axis	33.5%	15.5%	11.5%
y-axis	35.1%	23.9%	16.6%

## 7. Conclusions

The main goal of this study was to use the Kalman filter to improve the application of automotive LiDAR signals in severe weather. At present, LiDAR is used in many places, including in autonomous vehicles, but when the weather is bad, they are impossible to operate. More and more studies focus on how to improve LiDAR data in adverse weather. This paper used a combination of two methods to filter out undesired noise, first by using a nearest neighbor segmentation algorithm, and second by Kalman filtering. Both methods can be used separately, but they were combined here to filter out more noise. The nearest neighbor point cloud segmentation algorithm was used to filter out the noise of rain, and the Kalman filter was used to filter out the noise of dense smoke. The results of this study showed that signals measured in adverse weather can be restored to near normal data. The method described in this study can be used to obtain improvement, using the raw data for

comparison of normal weather with the bad weather data. The main objectives completed in this study are as follows:

- (1) This paper combined two algorithms to restore normal weather data more effectively.
- (2) Three simulated weather patterns were measured, and the LiDAR response observed.
- (3) The improvement rate obtained from the combination of the two algorithms is between 10% and 30%.

This research describes a LiDAR system capable of reducing environmental noise, which includes an optical radar and a processing module. The LiDAR obtains point cloud model data, and a processing module processes it from the self-detection environment. The processing module includes a nearest neighbor point cloud segmentation algorithm and a Kalman filtering unit. The nearest neighbor point cloud segmentation algorithm receives and processes the point cloud model data to output filtered model data, and the Kalman filtering unit receives the filtered model data to filter and output low-noise environment model data.

Looking to the future, the current measurements occurred in a controlled laboratory environment, and whether the system is controllable in real severe weather environments needs to be discussed via the following scenarios:

1. Use scientific instruments to measure severe weather data, raindrop size, and smoke concentration and particle size, for quantification of severe weather values.
2. Add different kinds of severe weather for testing and verification.
3. Conduct actual road tests in different adverse weather conditions.
4. All equipment used in this is vehicle-specific. Although they were tested in the laboratory, they can be installed on vehicles in the future.

If the above items can be thoroughly investigated, actual vehicle testing and verification will be carried out. If we can overcome the impact of bad weather on LiDAR, it will enhance the performance of self-driving vehicles and effectively improve human transportation.

**Author Contributions:** Conceptualization, S.-L.L.; methodology, S.-L.L.; software, B.-H.W.; validation, B.-H.W.; experiment, B.-H.W.; resources, S.-L.L.; data curation, B.-H.W.; writing—original draft preparation, B.-H.W.; writing—review and editing, S.-L.L.; visualization, S.-L.L.; supervision, S.-L.L.; project administration, S.-L.L.; funding acquisition, S.-L.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** The authors would like to thank the Ministry of Science and Technology, Taiwan, for financially supporting this research Grant no. MOST 109-2222-E-230-001-MY2.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Tran, Q.-D.; Bae, S.-H. An efficiency enhancing methodology for multiple autonomous vehicles in an Urban network adopting deep reinforcement learning. *Appl. Sci.* **2021**, *11*, 1514. [[CrossRef](#)]
2. Ma, X.; Hu, X.; Weber, T.; Schramm, D. Traffic simulation of future intelligent vehicles in duisburg city inner ring. *Appl. Sci.* **2021**, *11*, 29. [[CrossRef](#)]
3. Riedmaier, S.; Schneider, D.; Watzenig, D.; Diermeyer, F.; Schick, B. Model validation and scenario selection for virtual-based homology of automated vehicles. *Appl. Sci.* **2021**, *11*, 35. [[CrossRef](#)]
4. James, D.K. Stable analytical inversion solution for processing Lidar returns. *Appl. Opt.* **1981**, *20*, 211–220.
5. Rocadenbosch, F.; Reba, M.N.M.; Sicard, M.; Comerón, A. Practical analytical backscatter error bars for elastic one-component Lidar inversion algorithm. *Appl. Opt.* **2010**, *49*, 3380–3393. [[CrossRef](#)]
6. Mao, F.; Gong, W.; Ma, Y. Retrieving the aerosol Lidar ratio profile by combining ground- and space-based elastic Lidars. *Opt. Lett.* **2012**, *37*, 617–619.

7. Feiyue, M.; Wei, G.; Chen, L. Anti-noise algorithm of lidar data retrieval by combining the ensemble Kalman filter and the Fernald method. *Opt. Express* **2013**, *21*, 8287–8297.
8. Kutila, M.; Pyykönen, P.; Ritter, W.; Sawade, O.; Schäufele, B. Automotive Lidar Sensor Development Scenarios for Harsh Weather Conditions. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 265–270.
9. Kutila, M.; Pyykönen, P.; Holzhüter, H.; Colomb, M.; Duthon, P. Automotive LiDAR Performance Verification in Fog and Rain. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018.
10. Chen, J.; Ye, P.; Sun, Z. Pedestrian Detection and Tracking Based on 2D Lidar. In Proceedings of the 2019 6th International Conference on Systems and Informatics (ICSAI), Shanghai, China, 2–4 November 2019; pp. 421–426.
11. Schindler, P.; Heinzler, R.; Seekircher, J.; Ritter, W.; Stork, W. Weather Influence and Classification with Automotive Lidar Sensors. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 1527–1534.
12. Heinzler, E.; Piewak, F.; Schindler, P.; Stork, W. CNN-based Lidar Point Cloud De-Noising in Adverse Weather. In *IEEE Robotics and Automation Letters*; IEEE: Piscataway, NJ, USA, 2020; pp. 2514–2521.
13. Wallace, A.M.; Halimi, A.; Buller, G.S. Full Waveform Lidar for Adverse Weather Conditions. In *IEEE Transactions on Vehicular Technology*; IEEE: Piscataway, NJ, USA, 2020; pp. 7064–7077.
14. Jokela, M.; Kutila, M.; Pyykönen, P. Testing and Validation of Automotive Point-Cloud Sensors in Adverse Weather Conditions. *Appl. Sci.* **2019**, *9*, 2341. [[CrossRef](#)]
15. Hadj-Bachir, M.; Souza, P. *Lidar Sensor Simulation in Adverse Weather Condition for Driving Assistance Development*; ESI Group: Paris, France, 2019.
16. Zang, S.; Ding, M.; Smith, D.; Tyler, P.; Rakotoarivelo, T.; Kaafar, M.A. The Impact of Adverse Weather Conditions on Autonomous Vehicles: How Rain, Snow, Fog, and Hail Affect the Performance of a Self-Driving Car. In *IEEE Vehicular Technology Magazine*; IEEE: Piscataway, NJ, USA, 2019; pp. 103–111.
17. Zhou, Z.; Hua, D.; Wang, Y.; Yan, Q.; Li, S.; Li, Y.; Wang, H. Improvement of the signal to noise ratio of Lidar echo signal based on wavelet de-noising technique. *Opt. Lasers Eng.* **2013**, *51*, 961–966. [[CrossRef](#)]
18. Yoneda, K.; Suganuma, N.; Yanase, R.; Aldibaja, M. Automated driving recognition technologies for adverse weather conditions. *IATSS Res.* **2019**, *43*, 253–262. [[CrossRef](#)]
19. Duthon, P.; Colomb, M.; Bernardin, F. Light transmission in fog: The influence of wavelength on the extinction coefficient. *Appl. Sci.* **2019**, *9*, 2843. [[CrossRef](#)]
20. Naboulsi, M.A.; Sizun, H.; Formal, F. Fog attenuation prediction for optical and infrared waves. *Opt. Eng.* **2004**, *43*, 319–329. [[CrossRef](#)]
21. Kalman, R.E. A new approach to linear filtering and prediction problems. *J. Basic Eng.* **1960**, *82*, 35–45. [[CrossRef](#)]
22. Li, Q.; Li, R.; Ji, K.; Dai, W. Kalman Filter and its Application. In Proceedings of the 2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS), Tianjin, China, 1–3 November 2015; pp. 74–77.
23. Huang, W.; Li, Y.; Wen, P.; Wu, X. Algorithm for 3D Point Cloud Denoising. In Proceedings of the 2009 Third International Conference on Genetic and Evolutionary Computing, Guilin, China, 14–17 October 2009; pp. 574–577.
24. Kulawiak, M.; Lubniewski, Z. Processing of Lidar and Multibeam Sonar Point Cloud Data for 3D Surface and Object Shape Reconstruction. In Proceedings of the 2016 Baltic Geodetic Congress (BGC Geomatics), Gdansk, Poland, 2–4 June 2016; pp. 187–190.