

Vehicle Trajectory-Tracking Model for AV using LiDAR in Snowy Weather Under Different Snowing Environments

Padmapriya J¹, Ravi Prasath M² and Murugavalli S³

¹Panimalar Engineering College Chennai City Campus, Tamilnadu, India

²National Institute of Technology Tiruchirappalli, Tamilnadu, India

³Panimalar Engineering College Chennai City Campus, Tamilnadu, India

padmapriyaj.panimalar@gmail.com, 110121051@nitt.edu, principal@panimalarengineeringcollegechennai.ac.in

Abstract -

Autonomous cars are frequently fitted with radars, cameras, and LiDAR due to their complimentary capacities of environment awareness. However, when the host vehicle's LiDAR is negatively impacted by challenging weather circumstances like snow in varied snowfall conditions, precisely following the trajectory of the previous vehicle becomes imperative. Due to the sparse nature of LiDAR data, which can be influenced by a variety of variables such as the wind or snow conditions, it is also increasingly difficult to precisely remove the snow while maintaining the point clouds' details. The complete, learning-based strategy put out in this study attempts to address this urgent issue. Intensity and spatial-temporal feature-based de-snowing, QLGM, weighted technique, and switch RNN with a dual-level long short-term memory (LSTM) are used to track the trajectory path in snowy weather for various snowing levels.

Keywords -

De-noising; Trajectory Tracking; GMM; LiDAR; Autonomous Vehicle

1 Introduction and Background

A major challenge for fully autonomous military vehicles is navigating in bad weather since they lack the sensors to estimate depth, locate obstacles, and maintain traction. By getting troops out of high-risk environments such as snowstorms, autonomous vehicles can reduce casualties in the coldest places, such as Kargil in India, where the lowest temperature in winter can be as low as -35 degrees and the likelihood of contracting hypothermia is high. Because of the high level of precision in the ranging results, point clouds created by LiDAR could be a useful complement [1]. Environmental elements like snowflakes, fog, and raindrops cause noise in the point clouds [2],[3]. These airborne particles deflect laser beams, causing noise to appear in the corresponding locations in point clouds. However, studies on de-noising for snow have not been conducted [4],[5].

De-snowing for LiDAR, or fusing the de-noised point

clouds with the radar data stream, could help predict trajectory tracking under different snow conditions. Intensity is part of the LiDAR measurement outcomes [6], and this paper focuses on how to measure intensity in different types of snowfall. The intensity of snow points, which reflects the details of the environment, can be used to work out how much snow should be de-snowed [7]. One problem to be solved is how to use the intensity without harming the environment.

In accordance with the angles of incidence as well as Time of Flight (TOF), LiDAR points' locations could be calculated [8]. Snowflakes move in various directions and at various speeds, unlike raindrops. The motion of snow points is readily influenced by outside variables like wind and snowy conditions [9], [10].

The method for excluding snow-points from LiDAR data is presented in the paper. The central concept of our strategy is derived from two observations made in [11]. In temporal space, snow points are discrete, whereas the placement of non-snow points is typically constant. Our method first removes snow using an intensity-based filter that was based on the work in [12]. A restoring method is suggested to recover the lost points in the following stage of the process in order to enhance the accuracy of the point clouds. This paper proposes a de-snowing technique that combines the snow's intensity with the spatial-temporal characteristics of point clouds'. Additionally, it also proposes that the points in width and time (W-T) space, that can differentiate between snow points and point clouds, be restored.

Millimeter-wave (MMW) radars, LiDARs, and cameras are frequently found in fully autonomous vehicles (FAVs) in order to provide them with a driving experience that is as human-like as possible [13]. The motion capture of on-road users is the preliminary step in carrying out complex self-driving tasks. Precise tracking employing sensor fusion is a topic that both educators and corporate are very interested in. Spatial reasoning on complex data and the integration of multiple-projected paths, however, present two significant implementation challenges.

K-means [14] and the Gaussian mixture model (GMM)

[15] are two techniques that have been developed to group dense data in the existing literature on spatial reasoning. Thereinto, A complex object can be presented as a combination of a finite number of Gaussian density functions with unknown parameters, and GMM offers cutting-edge model-free technology for fitting it. To determine the maximum likelihood to estimate submodel parameters iteratively, the expectation-maximization (EM) algorithm is introduced [16].

It is possible to fit complex components with the Gaussian Mixture Model using cutting-edge model-free technology (GMM). A small amount of Gaussian density functions with uncertain attributes are combined to form the GMM. The expectation-maximization (EM) algorithm is used [5] to ascertain which submodel parameters have the highest probability of being estimated accurately. The two following significant drawbacks of the GMM-EM method are: 1) The quantity of clusters and 2) The probability that the submodel mean coincides with the values of real-world objects. If the centroids are not correctly reconfigured, the target vehicle's estimated trajectory will be off.

Motion traceability for the vehicle ahead becomes an important step in challenging snowy conditions, particularly when LiDAR information is disrupted by snow particles. It is important to move the GMM centroids which correlate with the velocity data. The integration of multi radar information is a critical issue that must be resolved in radar fusion.

Vision-based fusion frameworks have made it possible to detect and track multiple objects [17] [18]. However, because these algorithms rely on LiDAR information or camera streams, their motion predictions are unreliable. Recurrent neural networks (RNNs) exhibit a considerable advantage in designing dynamic systems and trajectories [19]. Furthermore, experiments have demonstrated the effectiveness of vehicle motion prediction using an RNN with LSTM cells. [20].

In previous works [21] - [22], methods for obtaining effective performance by sensor fusion were investigated. Saleh et al. proposed a three-layer LSTM network to track the trajectory of the pedestrian [23] and Yang et al. proposed a JTSM for customized trajectory prediction. [22]. Accurate trajectory tracking has been hindered by RADAR valid-range changes and LiDAR difficulties. The expansion of automated driving's operating environment and application space faces a significant challenge during snowfall. When it snows, the surrounding roads are in a different state than they are during fair weather. This is likely to be employed for tasks like operating snowplows that call for previous driving experience. One problem is that, due to snowfall and other factors, it is difficult to match the map data with the road surface. This makes it difficult to engage in safe automated driving because the lane a vehicle

is in can be hard to tell.

A number of studies have been conducted to address the issue by simulating the uncertainties associated with error propagation and afterwards comparing radar images [24]. In [25] LiDAR and millimetre wave radar were proposed as a localization system. Radar image alignment was incorrect because the confidence estimation was only done for LiDAR data, which resulted in inaccurate confidence level estimation. Additionally, it only takes into account the two environmental conditions of snow and no snow.

Therefore, a learning-based method has been proposed to track trajectories in the snow - covered climate under various snowing conditions. The three significant contributions are listed below: 1) the input LiDAR data stream is de-snowed using intensity and spatial-temporal features to separate snow points from point cloud data using spatial geometric data. 2) To acquire the precise positions of the clusters of compact short-range radar data, the amount of clusters is then investigated using a Q-learning method which is given to GMM-EM. The centroids are realigned in order to further mitigate the position deviation based on velocity inputs. (3) In addition, a switch dual-level LSTM (SDL) network approach is used that relies on the recommended radar-vision binary fusion technique to work with three snowing conditions. The vehicles that are ahead of them are tracked using SDL networks in challenging snowy conditions.

2 Approach

Fig. 1 depicts the proposed approach's framework. There are three steps in the framework. Prior to restoring non-noise, snow is first removed using an intensity filter and spatial-temporal features. The second step involves fusing the short-range radars (SRR) and electronically scanning radar (ESR) to assist in the prediction of trajectory tracking, which is done with the help of the de-snowed LiDAR points. And the final step is to use a Switch Dual-Level LSTM Network with combined filtered point clouds and fused radar data in order to predict the trajectory for three different modes.

2.1 De-snowing

2.1.1 Intensity-Based Filter

The ranging distance R , the incident angle, and the reflectance all play a significant role in intensity, according to the LiDAR principle in Eq. 1 [26].

$$I = H\left(\frac{P_t D_r^2 \rho \cos(\alpha)}{4R^2} \eta_{sys} \eta_{atm}\right) \quad (1)$$

Applying the settings in [27], the snowflake structure [28] and the LiDAR [11], α and the reflectance of snowflakes

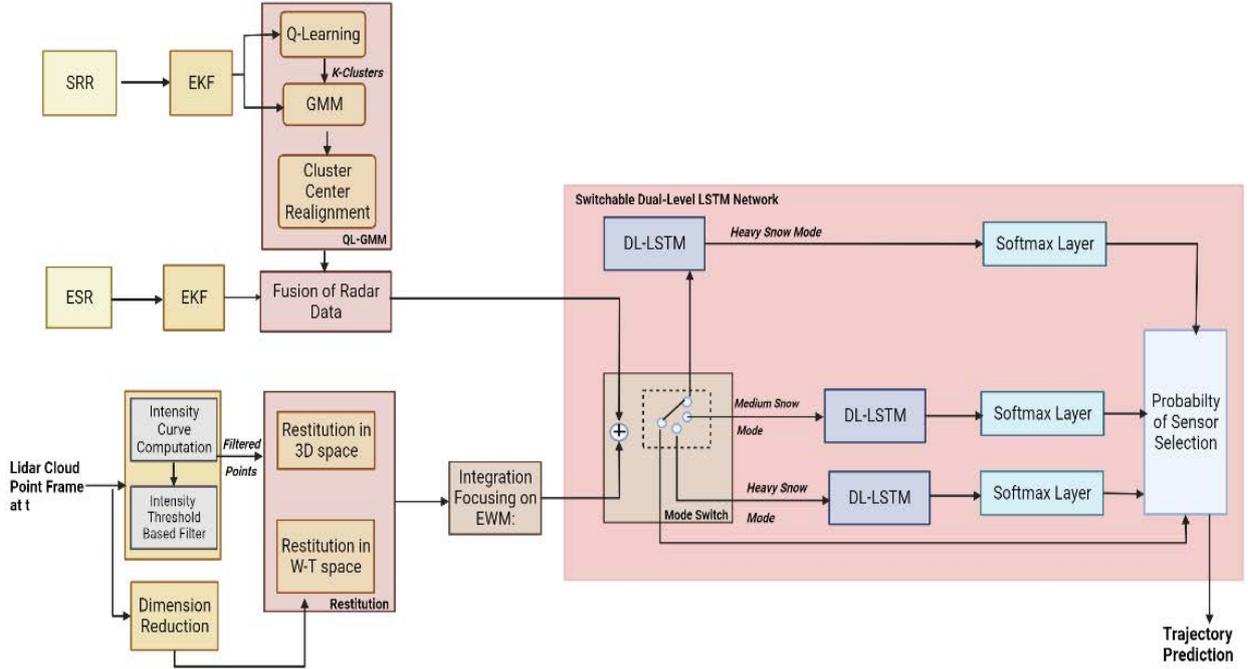


Figure 1. An Illustration of the proposed trajectory tracking framework.

are set . Using the measured intensity I_{ref} and the range R_{ref} , another intensity I_m with the range R_m could be calculated as in Eq. 2. In this manner , an intensity gradient of the snowflakes is obtained from several I_m at different R_m .

$$I_m = I_{ref} \frac{R_{ref}^2}{R_m^2} \quad (2)$$

2.1.2 Restitution

To recover the removed points with the same characteristics as snow points, spatial-temporal geometry features in typical 3D and width and time (W-T) space have been used. [11]. The amount of neighbour of a LiDAR point p_i in normal 3D space and W-T space, respectively, is denoted as symbols $|3d|_{p_i}$ and $|wt|_{p_i}$.

1) 3D space Restitution: Based on prior hypotheses, it is concluded in this paper that non-snow points are not isolated. A point is almost certain that it is a non-snow point if it has a lot of neighbours. The number of LiDAR points expressed as $p_i(x, y, z)$ for which Euclidean Distance to p_i is lower than r is known as $|3d|_{p_i}$ of p_i .

Because LiDAR is a sparse sensor, when r is set too low, there are not many neighbours in the distance. Distanced points will be taken into consideration as noise. The neighbour numbers resemble one another when r is set too high. Snow and point clouds are difficult to separate. r is set based on the experimental findings in [27].

2) W-T Space restitution: W-T space refers to the point of view whose centerline is parallel to the width and time axes . Images are viewed in W-T space using the height direction. As a result, from the frames, it was possible to view the movement of the pixels migrated at a particular height. Dimension reduction is used to reduce a collection of point cloud frames with a 4D tensor P to a 3D tensor P' . P' is regarded as a collection of 2D frames for the W-T space. When snow points are produced around the LiDAR, it is not feasible to have $|3d|_{p_i}$ alone to differentiate the snow. Since the trajectory of some weak-intensity points, like those from moving objects or the surface, is continuous, these points are grouped together in W-T space. As a result, it was possible to distinguish between non-snow points and weak- intensity points.

The amount of the nearest neighbour is used to specify the temporal feature given the constant trajectory and position of point clouds in W-T space. $p_i(w, h, t)$, in which w , h , and t denote the positions and stamp in the point clouds, respectively, can be used to represent a LiDAR point in W-T space. From time $t - n/2$ to time $t + n/2$, the number n indicates how many point clouds are in the area. The amount of points in a $(n + 1) \times (n + 1)$ searching mask in W-T space is indicated by the symbol $|wt|_{p_i}$ of p_i at height h .

Prior to computing $|wt|_{p_i}$, point clouds are preprocessed. In the W-T space, $|wt|_{p_i}$ is computed. Each point's 3D coordinates (x , y , and z) cannot be used directly. The Principal Component Analysis (PCA) is used to transform 3D

point clouds into 2D [29].

3) Integration Focusing on EWM: Entropy is an indicator of system disorder, based on information theory. It is used to determine the degree of discreteness. The degree of discreteness increases with decreasing entropy value, and this has a greater impact primarily on extensive assessment. As a result, entropy may be used to compute weights for various indicator models. The entropy weight method (EWM) is a weight computation method that produces results according to the data dispersion.

In this case, $|3D|_{pi}$ and $|wt|_{pi}$ will be combined to calculate a weighted average confidence score to decide if the point must be recovered [11]. As a result, EWM is used to compute the weights. P_i 's calculated $|3D|_{pi}$ as well as $|wt|_{pi}$ is saved in $F_{m \times 2}$. Weights w_j , distribution of $|3D|_{pi}$ and $|wt|_{pi}$ expressed as pro_{ij} , and entropy values e_j are calculated using EWM as in [11]. After calculating w_1 and w_2 , Eq. 3 computes a weighted Score I of pi using the normalised $|3D|_{pi}$ and $|wt|_{pi}$. Score i is the confidence score used to decide if pi is the snow point when compared to a threshold. The threshold is set as mentioned in [11].

$$Score_i = w_1 \cdot \frac{|3D|_{pi}}{|3D|_{pi} + 1} + w_2 \cdot \frac{|wt|_{pi}}{|wt|_{pi} + 1} \quad (3)$$

2.2 Fusion of Radar Data

The Extended Kalman Filter (EKF)-filtered data from SRRs is given to the Q-learning GMM clustering component. To produce the fused radar stream Φ_R^R , EKF-processed SSR data is paired with the EKF-processed ESR data. Φ_R^R and de-snowed Lidar Cloud Points Φ_D^D are then combined to form Φ_{RD}^{RD} . It is sent to the snowing condition switch, a finite state machine (FSM). A dual-level LSTM network layer and a softmax layer are coordinated by each snowing mode. Following that, the sensor fusion block and network blocks can be used to achieve the predicted trajectory.

2.2.1 GMM for Clustering Based on Q-Learning

The standard GMM-EM algorithm is explained and illustrated in [16], [23], and [22], and it can be written as

$$p(\xi_{ip}|\theta) = \sum_{k=1}^K \rho_k \mathcal{N}_k(\xi_{ip}|\Sigma_k) \quad (4)$$

where $\xi_{ip} = (X_{SRR_{ip}}, Y_{SRR_{ip}})$ as well as $\Theta = \{\rho_k, \mu_k, \Sigma_k\}_{k=1}^K$ are model parameters and the position data sequence. The coordinates' index is i_p , and the length of the sequence is indicated by n_p . Gaussian submodels are represented by the number K . Multivariate Gaussian distribution is denoted by $N_K(\xi_{ip}|\mu_k, \Sigma_k)$, where Σ_k

and μ_k are the covariance matrix and mean vector. The k^{th} Gaussian sub model that satisfies has a component weight of $0 < \rho_k \leq 1$, and this value is represented by. However, there is one significant flaw with plain K-means or traditional GMM. That is to say, it is unknown how many clusters K there will be.

K was specified in advance as a constant for the duration of the implementation in the literature [15] and [22]. However, given the ambiguity and time-varying nature of driving conditions, it is irrational to assume that the surroundings can be divided. Noise and initialization can have a significant impact on clustering performance, even though K can be automatically determined by Bayesian non parametric [16].

To this end, it is possible to develop a Q-learning-based strategy to figure out K in this section [30]. In addition to offering an offline Q-value table for implementation at a lower cost in terms of computation, Q-learning can also maintain good performance even when clustered targets are close together. QLGM is also unaffected by initial conditions, noise in the data, and outliers.

2.2.2 GMM Algorithm Based on Q-Learning

A commonly applied model selection criterion, BIC [16], [21],[31], can be employed to determine the amount of Gaussian components K . The BIC function, denoted as $g(\hat{L})$, is given by

$$g(\hat{L}) = BIC = -2 \ln(\hat{L}) + M_p \ln(M) \quad (5)$$

For Q-learning-based selection of K , a Markov decision process (MDP) is presented [32]. Given that action space A , the amount of clusters K with K being the finite number of model components, the BIC results necessitates the log-likelihood estimation \hat{L} feedback. The prior likelihood of the chosen course of action is represented by P_a . A one-to-one correspondence can be established as a result to increase the effectiveness of training [33].

The state-action integration Q is used by Q-learning to derive a Q-value table [34]. Every Q-value table item is initially set to $Q_0 = \frac{K_0}{BIC_0}$ rather than 0, reducing the iterative computation. K_0 is a positive weight, and BIC_0 stands for the previously calculated BIC value at various K .

Using the Q-value component the result of carrying out action at in state s_t while governed by π at time interval τ_s is denoted by the symbol $Q_\pi(s_t, a_t)$, which is used to serve the Q-value table. This value is calculated by starting in state s_{τ_s} , carrying out action a_{τ_s} , and then carrying out policy π . The Q-value table can be constantly monitored from the ω^{th} iteration to the $(\omega + 1)^{th}$ iteration by adopting a learning rate of $(0 < \alpha \leq 1)$ and the temporal

difference in [34].

The quantity of radar-data clusters in the T-frame scenario can be determined after the optimal policy π^* is achieved. The position data for each cluster is then determined by feeding π^* into the standard GMM-EM algorithm. In contrast to using a constant [15] or the BIC traversal calculation [16], Therefore, it is necessary to determine how many clusters are present in each frame [15]. This method can also offer a novelty to choose the best course of action for time-series LiDAR data during a driving scenario without losing generality [30].

2.2.3 Cluster Center Realignment

The Gaussian sub-models don't take into account the velocity information v_k^{SRR} ($v_{x|k}$, $v_{y|k}$) as a potential centering mismatch cause [31]. Furthermore, it is unclear how much each vehicle weighs. So the cluster centres realigned using a kinetic-energy-element method. Each point's kinetic energy per unit mass (KEUM) is represented by the notation $\tilde{E}_{kin|k_i k}$ and its relocated centre by the notation $\tilde{\zeta}_k^{SRR}$. Because of the introduction of KEUM, the re-alignment results now correspond to the actual vehicle positions. By using this method, the suggested SRR cluster centres could be more effectively linked to the ESR data.

2.2.4 Radar Data Association

The suggested QLGM method was used to cluster the SRR results in the previous section. But another problem that needs to be solved is how to combine the data from multiple radars.

In addition to these other data, the acquired radar data streams textPhi_T^{SRR} and Φ_T^{ESR} contain position coordinates ξ , velocity signals v , relative distance D , and the direction Ψ of the vehicle ahead. The integrated T-frame radar stream RT is created by combining the reconfigured QLGM SRR data $\text{textPhi}_{QLGM}^{SRR}$ as well as the EKF ESR data Φ_{EKF}^{ESR} . which is given as

$$\Phi_T^R = \lambda_1 \Phi_T^{SRR} + \lambda_2 \Phi_T^{ESR} \quad (6)$$

The proximity of a target to the radar system and the amount of radar points on a vehicle both affect how well a vehicle detects objects with radar [35]. Due to the unknown designed fusion weights of SRR (λ_1), detection-area-based method is addressed focused on the radar specifications as well as fitting boundaries [30]. The detection area is divided into five sections as in [30].

In a data-driven manner, λ_1 can be developed as a hybrid linear function in accordance with the fitting boundaries. As a result, the filtered LiDAR data stream and fused radar data are combined as Φ_T^{RD} and given to the SDL networks for trajectory prediction.

2.3 LiDAR-Fault-Tolerant Trajectory Prediction

2.3.1 FSM for Snowing Mode Switch

In this paper, the signals of the de-snowed LiDAR point clouds can be used to determine three snowing modes: heavy snow mode, medium snow mode, and light snow mode. FSM [36] [37] is created to change between various snowing modes. The swapping logic and the Snowing modes are represented by three states and six transitional conditions in this FSM, similar to [30]. The $I_{th1} > I_{th2}$ and $0 < I_{th1} < I_{th2}$ conditions are met by the thresholds of snowing intensity for state transition. Depending on how much snow is falling, the FSM holds each state or switches to another. As long as the snowfall intensity stays within a certain range, the current snowing mode would be retained. Otherwise, upon meeting or exceeding the thresholds, the state transition is enabled. As defined below, the FSM's switching actions include:

- 1 Maintain the Light Snow mode: $I \geq I_{th1}$.
- 2 Swap between Light Snow and the Medium Snow mode: $|I| \leq I_{th1}$.
- 3 Maintain the Medium Snow mode: $I_{th2} < I < I_{th1}$.
- 4 Swap between Medium Snow and the Heavy Snow mode: $I_{th1} < |I| < I_{th2}$.
- 5 Maintain the Heavy Snow mode: $I \leq I_{th2}$.
- 6 Swap between Heavy Snow and the Light Snow mode: $|I| \geq I_{th2}$.

2.3.2 Dual-Level LSTM Network for Trajectory Prediction

The RNN with LSTM cells enable a workable alternative to estimate the future trajectory because of their improved ability of capturing long temporal correlations [37]. The performance of a single LSTM layer, however, is sub-par in the actual implementation [38]. Additionally, an excessive number of LSTM layers will cause the computational load to rise exponentially [39]. The hidden state h between LSTM cells and then a softmax probability vector y are created by each snowing mode coordinating a dual-level-LSTM layer and a softmax layer [30]. The LSTM (.) operator is used as the LSTM to prevent gradient vanishing.

The prediction accuracy is further improved by a cooperative dynamic binary fusion approach by including the SDL layers [30]. The binary sensor choice following the softmax operation should be represented by a block matrix $P_\tau = [I_\tau, O_\tau], [O_\tau, I_\tau], I_\tau, O_\tau \in \mathbb{R}$, where O_τ and I_τ are the zero matrix and the identity matrix at a time step, respectively. $y_\tau = P_\tau (1) = [I_\tau, O_\tau]$ indicates using radar data, and $y_\tau = P_\tau (2) = [O_\tau, I_\tau]$ indicates choosing de-snowed LiDAR data. The foremost sources to the suggested SDL network are orientation, relative distance

and longitudinal velocity signals from LiDAR and radar streams. The results of selection probability indicate "1" for radars only whereas "0" for only de-snowed LiDAR.

2.4 Light Snow Mode

Due to the sparse snow or clear weather in the Light Snow mode, the result of the proposed method would be from LiDAR data. In this mode the LiDAR point clouds will not have noise from the snow as the amount of snow will be less or none, therefore the point clouds from the de-snowed LiDAR can be used to predict the trajectory tracking.

2.5 Medium Snow Mode

The outcome of the suggested approach might alter between fused radar data and the de-snowed LiDAR method in medium snowing condition as the intensity may vary from low to high and the LiDAR data might have more noise. Therefore, the proposed QL-GMM+SDL method is the optimal solution for the given mode.

2.6 Heavy Snow Mode

In this condition with de-snowed LiDAR data, there is the possibility of two types of detection failures: miss detection and mishap. Detecting objects is more challenging when using de-snowed LiDAR data in this mode. In this initialization region, the radar stream is a necessary component of the proposed data fusion.

3 Conclusion

Navigating inclement weather presents a significant challenge for fully autonomous vehicles. Snow particularly confounds pivotal sensor data that a vehicle needs to gauge depth, find obstacles, and maintain traction. For autonomous vehicles operating in a variety of snowy circumstances, a de-snowing method for LiDAR point clouds and an integrated learning-based vehicle tracking solution is suggested in this paper. The spatial-temporal features retrieved from 3D and W-T space are used to filter out snow points and restore non-snow points. This solution includes a weight scheduling approach for multi-radar integration, an SDL network for trajectory tracking in three snowfall conditions, and a Q-learning-based GMM-EM algorithm for compact data aggregation. One of the two promising enhancements of the QLGM algorithm is the statistical determination of the number of clusters by Q-learning. Suggested re-alignment concepts, in contrast, can eliminate the positional incompatibilities between the GMM cluster centres and actual automobiles. Additionally, to create a fused radar data stream, QLGM SRR cluster centres can be linked to ESR data

based on the EKF. To improve the accuracy of vehicle tracking, an FSM-based SDL network model with a binary fusion method that matches the different snowfall conditions is designed. The proposed method ingeniously enhances the radar-vision tracking efficiency when faced with different environmental conditions, such as snowy weather in difficult snowy circumstances, making it a promising replacement for humans in dangerous military operations.

The results of the simulation model and evaluation of the proposed method will be compared and analyzed in the future. The emphasis of upcoming work will be on implementing suggested algorithms in real-time. In conditions with snowfall, it may not be possible to match the pattern of the road surface, so a robust system must reject matching results or use additional sensors to make up for them. There this idea will be further extended to have an integrated multi-sensor fusion Vehicle Localization and Trajectory Tracking Framework under different snow conditions.

References

- [1] Y. Zhou and O. Tuzel. "voxelnet: End-to-end learning for point cloud based 3d object detection". In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, page 4490–4499, 2018.
- [2] M. Colomb Y. Li P. Duthon and J. Ibanez-Guzman. "what happens for a tof lidar in fog?". *IEEE Trans. Intell. Transp. Syst.*, page 1–12, 2020.
- [3] M. Byeon and S. W. Yoon. "analysis of automotive lidar sensor model considering scattering effects in regional rain environments". In *IEEE Access*, page 102669–102679, 2020.
- [4] J. Shi W. Xu Y. Wang F. Fu and J. Wang. "efficient moving objects detection by lidar for rain removal". In *Int. Conf. Intell. Comput. (ICIC)*, page 697–706, 2016.
- [5] A. U. Shamsudin K. Ohno T. Westfechtel S. Takahiro Y. Okada and S. Tadokoro. Fog removal using laser beam penetration, laser intensity, and geometrical features for 3d measurements in fog-filled room". *Advanced Robotics*, 30(11-12):729–743, 2016.
- [6] P. Vacek O. Jaek K. Zimmermann and T. Svoboda. "learning to predict lidar intensities". *IEEE Trans. on Intell. Transp. Syst.*, page 1–9, 2021.
- [7] S. Ronnback and A. Wernersson. "on filtering of laser range data in snowfall". In *Proc. Int. Conf. Intell. Syst.*, volume 2, page 17–33, 2008.

- [8] J. Liu Q. Sun Z. Fan and Y. Jia. “tof lidar development in autonomous vehicle”. In *Proc. IEEE Optoelectron. Global Conf. (OGC)*, page 185–190, 2018.
- [9] B. Yang Z. Jia J. Yang and N. K. Kasabov. “video snow removal based on self-adaptation snow detection and patch-based gaussian mixture model”. *IEEE Access*, 8:160188–160201, 2020.
- [10] C. H. Bahnsen and T. B. Moeslund. “rain removal in traffic surveillance: Does it matter?”. *IEEE Trans. Intell. Transp. Syst.*, 20(8):2802–2819, 2019.
- [11] B. Li J. Li G. Chen H. Wu and K. Huang. “de-snowing lidar point clouds with intensity and spatial-temporal features”. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2359–2365, 2022. doi:10.1109/ICRA46639.2022.9812241.
- [12] J. I. Park J. Park and K. S. Kim. “fast and accurate desnowing algorithm for lidar point clouds”. In *IEEE Access*, volume 8, page 160202–160212, 2020.
- [13] M. Cao and J. Wang. “obstacle detection for autonomous driving vehicle with multi-lidar sensor fusion,”. *J. Dyn. Syst., Meas., Control*, 142(2), 2020.
- [14] J. Melo A. Naftel A. Bernardino and J. Victor. “detection and classification of highway lanes using vehicle motion trajectories. *IEEE Trans. Intell. Transp. Syst.*, 7(2):188–200, May. 2006.
- [15] W. Wang J. Xi and D. Zhao. “learning and inferring a driver’s braking action in car-following scenarios”. *IEEE Trans. Veh. Technol.*, 67(5):3887–3899, May 2018.
- [16] W. Wang D. Zhao W. Han and J. Xi. “a learning-based approach for lane departure warning systems with a personalized driver model”. *IEEE Trans. Veh. Technol.*, 67(10):9145–9157, Oct. 2018.
- [17] H. Gao B. Cheng J. Wang K. Li J. Zhao and D. Li. “object classification using cnn-based fusion of vision and lidar in autonomous vehicle environment,”. *IEEE Trans. Ind. Informat.*, 14(9):4224–4231, Sep. 2018.
- [18] D. Kim B. Kim T. Chuang and K. Yi. “lane-level localization using an avm camera for an automated driving vehicle in urban environments,”. *IEEE/ASME Trans. Mechatronics*, 22:280–290, Feb. 2017.
- [19] S. Xie and J. Ren. “recurrent-neural-network-based predictive control of piezo actuators for trajectory tracking,”. *IEEE/ASME Trans. Mechatronics*, 6: 2885–2896, Dec. 2019.
- [20] A. Zyner S. Worrall and E. Nebot. “naturalistic driver intention and path prediction using recurrent neural networks,”. *IEEE Trans. Intell. Transp. Syst.*, 21(4): 1584–1594, Apr. 2020.
- [21] K. Saleh M. Hossny and S. Nahavandi. “intent prediction of pedestrians via motion trajectories using stacked recurrent neural networks”. *IEEE Trans. Intell. Veh.*, 3(4):414–424, Dec. 2018.
- [22] X. Li W. Wang and M. Roetting. “estimating driver’s lane-change intent considering driving style and contextual traffic”. *IEEE Trans. Intell. Transp. Syst.*, 20(9):3258–3271, Sep. 2019.
- [23] W. Wang J. Xi and J. K. Hedrick. “a learning-based personalized driver model using bounded generalized gaussian mixture models”. *IEEE Trans. Veh. Technol.*, 68(12):11679–11690, Dec. 2019.
- [24] K. Yoneda N. Hashimoto R. Yanase M. Aldibaja and N. Sukanuma. “vehicle localization using 76 ghz omnidirectional millimeter-wave radar for winter automated driving,”. In *Proc. IEEE Intell. Vehicles Symp. (IV)*, page 971–977, Jun. 2018.
- [25] N. Yanase R. Hirano D. Aldibaja M. Yoneda, K. Sukanuma. “lidar- and radar-based robust vehicle localization with confidence estimation of matching results”. *Sensors*, 22:3545, 2022.
- [26] S. Kaasalainen A. Jaakkola M. Kaasalainen A. Krooks and A. Kukko. “analysis of incidence angle and distance effects on terrestrial laser scanner intensity: Search for correction methods”. *Remote Sens.*, 3(10):2207–2221, 2011.
- [27] J. I. Park J. Park and K. S. Kim. “fast and accurate desnowing algorithm for lidar point clouds”. *IEEE Access*, 8:160202–160212, 2019.
- [28] Su-Hyun Kim Dae-Hong Ko Dae-Kyung Seong Seung-Hee Eun Byung-Gon Kim Baek-Jo Kim Chang-Geun Park and Ju-Wan Cha. “quantitative analysis of snow particles using a multi-angle snowflake camera in the yeongdong region”. *Atmosphere*, 29(3):311–324, 2019.
- [29] Y. Duan C. Yang H. Chen W. Yan and H. Li. “low-complexity point cloud denoising for lidar by pca-based dimension reduction”. *Opt. Commun.*, 482: 26567, 2021.
- [30] N. Chen M. Cao R. Wang and J. Wang. “a learning-based vehicle trajectory-tracking approach for autonomous vehicles with lidar failure under various lighting conditions”. *IEEE/ASME Transactions*

on *Mechatronics*, 27(2):1011–1022, April 2022.
doi:10.1109/TMECH.2021.3077388.

- [31] P. J. Huber. “robust statistics,”. *International Encyclopedia of Statistical Science*. Berlin, Germany: Springer, page 1248–1251, 2011.
- [32] H. Ko S. Pack and V. C. Leung. “mobility-aware vehicle-to-grid control algorithm in microgrids”. *IEEE Trans. Intell. Transp. Syst.*, 19(7):2165–2174, Jul. 2018.
- [33] T. Back. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford Univ. Press, London, U.K., 1998.
- [34] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA, 2018.
- [35] H. Sun J. Xin X. Wang L. Xu and N. Zheng. “on-road vehicle detection and tracking using mmw radar and monovision fusion”. *IEEE Trans. Intell. Transp. Syst.*, 17(7):2075–2084, Jul. 2016.
- [36] Y. Chen J. Zha and J. Wang. “an autonomous t-intersection driving strategy considering oncoming vehicles based on connected vehicle technology,”. *IEEE/ASME Trans. Mechatronics*, 24(6): 2779–2790, Dec. 2019.
- [37] Y. Chen C. Hu and J. Wang. Human-centered trajectory tracking control for autonomous vehicles with driver cut-in behavior prediction,”. *IEEE Trans. Veh. Technol.*, 68(9):8461–8471, Sep. 2019.
- [38] L. Zhang R. Gao J. Wang P. Fu and R. Zhao. “multilevel information fusion for induction motor fault diagnosis”. *IEEE/ASME Trans. Mechatronics*, 24(5): 2139–2150, Oct. 2019.
- [39] X. Yang C. Lv and D. Cao. “personalized vehicle trajectory prediction based on joint time series modeling for connected vehicles”. *IEEE Trans. Veh. Technol.*, 69(2):1341–1352, Feb. 2020.