Traffic & Transportation

ICP-LSTM Joint Localisation for Autonomous Driving

Junhe LIU[1]

Original Scientific Paper
Submitted: 30 Apr 2025
Accepted: 12 Sep 2025
Published: 29 June 2026

[1] Corresponding author, junhe_liu@outlook.com, School of Mechanical and Power Engineering, East China University of Science and Technology, Shanghai, China

Publisher:
Faculty of Transport and Traffic Sciences, University of Zagreb

ABSTRACT
Autonomous driving requires precise vehicle localisation, especially in moving, unstructured environments. Existing methods focus on spatial matching or temporal sequence modelling for sensor noise, occlusions and partial data. This causes errors. Spatial-only methods struggle with environmental uncertainty, while temporal-only models lack geometric coherence. This study proposes the iterative closest point and long short-term memory-based joint localisation algorithm (ICP-LSTM-JLA) to solve these limitations. This hybrid approach combines geometric point cloud registration and learning-based motion sequence modelling. LSTM-driven predictions that employ inertial and motion data modify the system's initial posture estimation. Aligning real-time sensor data with pre-mapped environments does this. ICP-LSTM-JLA can enhance localisation accuracy and resilience in diverse driving conditions by integrating spatial alignment with temporal dynamics. The experiments showed that compared to state-of-the-art models, the absolute trajectory error (ATE) has decreased by 17.8%, the relative positioning error (RPE) by 14.3%, the map registration error by 19.1% and the orientation error by 12.6%. This study found that the hybrid architecture improves short-term stability and long-term trajectory accuracy. Thus, it is a reliable real-world autonomous navigation system.

KEYWORDS
autonomous vehicle localisation; point cloud registration; deep learning for navigation; LSTM-based motion prediction; sensor fusion in robotics; SLAM (simultaneous localisation and mapping); real-time navigation systems.

1. INTRODUCTION

Autonomous driving is expected to revolutionise transportation by reducing traffic, enhancing safety and increasing efficiency [1]. A reliable localisation system is necessary for autonomous vehicles (AVs). Localisation involves placing a vehicle in a global or local reference frame and orienting it according to its six degrees of freedom [2-3]. Data are required for advanced operations, such as control, mapping, obstacle avoidance and motion planning. As noted in [4], classical localisation usually involves inertial, optical and GNSS measurements. GNSS signals can be unstable or inaccessible in urban canyons, tunnels and other densely populated locations [5]. This has made enhanced sensor-based localisation technologies, such as IMUs, radar, monocular and stereo cameras, and LiDAR, popular [6]. LiDAR technology's ability to send three-dimensional spatial data in various illumination and weather conditions has made it popular [7]. Geometric registration methods, especially the ICP algorithm, are widely used for LiDAR-based localisation [8]. ICP selects the optimum transformation to align an input point cloud with a pre-mapped reference frame or a scanned object, minimising a point-to-plane or point-to-point error metric [9]. Static and ordered surroundings work well for this strategy. However, ICP is susceptible to outliers, dynamic barriers and geometric feature sparsity [10], requiring an accurate early pose estimate to converge to an ideal solution. ICP can converge to local minima in dynamic or feature-poor environments, resulting in erroneous pose estimation.

Since the ICP is deterministic and does not employ temporal information, its adaptability is limited to a large range of motion patterns [11].

The ICP methodology and other geometric registration methods are used for LiDAR-based localisation [12]. ICP's ideal transformation, which uses a point-to-plane or point-to-point error measure, aligns point clouds with pre-mapped reference frames or scans faster [13]. This method worked in controlled situations. Due to its susceptibility to outliers, dynamic barriers and geometric feature sparsity, a valid starting vehicle pose estimate is needed to find a solution. If the ICP converges to local minima under dynamic or feature-poor settings, pose estimation may be erroneous. ICP does not utilise temporal information and is deterministic, which limits its application to many motion patterns [14].

Most AV localisation system challenges stem from the lack of coherent frameworks that integrate geometric spatial alignment with the momentum dynamics of learning-based temporal models. Most existing techniques separate these zones, making it difficult to attain maximum performance in real-world and dynamic driving circumstances. Hybrid localisation frameworks that integrate ambient geometric components with sequentially learned motion contexts are now needed. This paper presents the joint localisation algorithm to overcome this problem. This method would combine the best ICP and LSTM networks, and the described system has two main components. The first part is an ICP-based geometric alignment module. This module uses frame-to-frame or frame-to-map registration to estimate pose from unprocessed LiDAR data. The second component, LSTM network training, utilises odometry and IMU data sequences, and this network learns the vehicle's movement. After that, this network creates a temporal inference-based vehicle tilt correction. Adjustable fusion employs a learnable gating network to dynamically select module contributions or weighted averages based on uncertainty predictions, thereby fusing these two outputs. Both approaches are adaptive combination methods. LSTM-only systems accrue drift; hence, the ICP-LSTM-JLA was created. This strengthened the ICP's resilience to unwanted initialisations and dynamic environmental changes. This comprehensive technique enhances reliability in complex urban environments by enabling precise and continuous localisation over time.

To enable safe navigation, decision-making and control in varied surroundings, autonomous driving systems rely on accurate and strong vehicle localisation. Occlusions, sensor noise, dynamic obstacles and insufficient map data can reduce localisation accuracy when driving in real-life scenarios. Older point cloud registration methods employ geometric spatial alignment techniques, such as ICP. Inertial and odometry data are used in temporal sequence modelling, another popular method. These solutions are useful yet have drawbacks. Temporal-only models wander because they lack a geometric foundation. Spatial-only methods struggle with sparse features and environmental uncertainty. The ICP-LSTM-based Joint Localisation Algorithm is presented in this article. This framework uses the strengths of the two methodological techniques to address these issues. The technology uses geometric point cloud alignment to understand the user's location. It then uses motion and inertial data from an LSTM-based motion prediction module to make a more precise assessment. ICP-LSTM-JLA, which incorporates spatial and temporal information, may help autonomous vehicles in complex, unstructured environments improve accuracy, resilience and adaptability. This study aims to develop a combined localisation technique that outperforms current state-of-the-art models in ATE, RPE, and map registration discrepancies. The evaluation and development of such an approach are the primary goals of this project. This paper demonstrates that the proposed technique can enhance the short-term stability and long-term trajectory accuracy of autonomous navigation in real-world applications through rigorous testing.

The main contributions of this research are summarised as follows:

1) Introducing a novel framework that combines the spatial accuracy of ICP with the temporal modelling capabilities of LSTM to achieve improved localisation performance in autonomous driving scenarios.
2) Developing a modular pipeline that integrates LiDAR, IMU, and odometry data in a parallel and real-time processing setup, supporting sensor redundancy and improved fault tolerance.
3) A customised LSTM network is trained to learn motion patterns from historical sequences and generate pose refinements fused with ICP outputs to reduce trajectory drift.
4) The proposed joint localisation algorithm (JLA) effectively combines geometric consistency with learned temporal dependencies to refine localisation outputs, enabling precise, drift-minimised positioning in dynamic driving environments.
5) The results demonstrate significant improvements in pose accuracy, with reductions in error rate compared to LSTM-only baselines.

This research is structured as follows: Section 2 reviews related work on LiDAR-based and learning-based localisation techniques. Section 3 describes the architecture of the proposed ICP-LSTM-JLA framework.

Section 4 presents the dataset, evaluation metrics and experimental results. Section 5 concludes the paper and suggests directions for future study.

2. LITERATURE SURVEY

Dai et al. [15] suggested the localisation of autonomous driving vehicles in complex scenarios using LiDAR-based sensor fusion, simultaneous localisation and mapping (SLAM). This research presents and integrates an SLAM and online localisation approach based on multi-sensor fusion into a generic framework. The suggested strategy is tested using field testing and the open-source KITTI dataset. It achieves online autonomous driving in complex situations, with a mapping accuracy of 5-10 cm and a localisation accuracy of 20-30 cm, as demonstrated by the test results.

Charroud et al. [16] proposed the modified clustering particle filter (MCPF) for fast and accurate localisation and mapping methods for self-driving vehicles. Based on LiDAR data, the design contains two primary steps: localisation and mapping. In the first step, mapping, the author employs a fuzzy c-means algorithm to generate global maps of the environment, utilising non-semantic features rather than the semantic algorithm used in existing works. The DBSCAN technique was used to filter out the other clusters. This research used a particle filter architecture for the localisation phase, which included resampling, motion updates and measurement updates to estimate locations. This research's most significant contribution is the unique extension of particle selection that lowers computation time while maintaining long-term localisation. Pandaset was widely utilised to test the chosen strategy. The author tested the method on a large sequencing dataset and compared it to other methods. The investigation found that the approach is fast and can acquire all the data needed for real-time localisation.

Xie et al. [17] suggested automatically labelling autonomous driving systems using weakly supervised object localisation (WSOL), a soft guiding module (SGM) and a channel erasing module. This study suggests using a multi-flow architecture to construct the channel erasing module (CEM) and soft guidance module (SGM), which may benefit both. In conclusion, the author tests the technique using the Stanford Cars, CUB-200-2011 and ILSVRC 2016 datasets.

Integrating a memory transformer with a double deep Q-network (DDQN), Zha et al. [18] suggested a multi-modal data fusion technique for autonomous cars' real-time localisation and navigation procedures. Using a data fusion technique that incorporates a self-attention network and the memory transformer for efficient data encoding, the model strikes a balance between algorithmic real-time speed and feature fidelity. Additionally, a DDQN vehicle navigation algorithm is employed, utilising encoded data that leverages large-scale scene maps and an automatically growing environmental target knowledge network. In practical applications, this allows for optimisation and ongoing learning. When it comes to utilising large-scale scene maps for macro-spatial reference, the automatically growing knowledge graph for background knowledge support, and the DDQN algorithm's experience-optimised navigation strategies, experimental comparisons show that the proposed model outperforms current SOTA models. When compared to the SOTA models, the following six metrics, NE, SR, OSR, SPL, CLS and DTW yielded results of 3.99, 0.65, 0.67, 0.65 and 0.63, respectively.

Cheng et al. [19] presented the map-aided visual-inertial fusion localisation technique for autonomous vehicles. The first step is to use the camera to capture and save a map of the low-speed area. The second run uses the loaded and matched map from the first run to estimate the vehicle's pose. The IMU helps correct visual mistakes while building the map. Experimental results demonstrate that the proposed strategy is highly effective for continuous localisation. The findings were based on the KITTI dataset and real-world conditions.

Liu et al [20] proposed positioning possibilities for urban autonomous driving using GNSS, LiDAR and 3D maps. 3D map-based global localisation (MGL) provides absolute positioning solutions on 3D maps. This method matches LiDAR images to maps. The recommended drift error correction solution reduces the cumulative error in LiDAR odometry by 35.5% compared to the current GNSS/LO fusion method.

Grebner et al. [21] discussed using SLAM to interpret synthetic aperture radar for autonomous driving. The author proposes a radar-only SLAM approach for high-resolution SAR processing at the 77 GHz automotive frequency. This effort produced this approach. The technique was tested by measuring 500-metre trajectories for over two minutes.

To help autonomous vehicles navigate complex settings, Syed et al. [22] proposed the YOLOv8n architecture, which utilises the convolutional block attention module (CBAM) to identify obstacles and estimate distances in real-time. Soft DIou-NMS enhances the non-maximum suppression (NMS) technique by

decreasing the frequency of item overlap detection. The performance was evaluated using a thorough dataset that included common orchard barriers. Confusion matrices, F1-score, precision (P) and recall (R) were among the measures utilised. Experimental results show that compared to various baseline models, including YOLOv5s and other variants of YOLOv8 (n, s, m, l and x), the YOLOv8n model performs better. A mean average precision (mAP) of 92.7% was attained at IoU 0.5, and an F1-score of 87% was attained at a confidence threshold of 42.1%. The inference time was reduced to 3.3 milliseconds, and the model size was reduced to 20.1 megabytes. The model achieved mAP-50 scores of 96.5% in bright light and 91.4% in dim light during evaluation, indicating slight performance declines in both environments.

Hu et al. [23] investigated adaptive false data injection and security analysis for multi-sensor fusion localisation in autonomous driving. Attacks using this strategy are more difficult to detect because they are stealthy. In contrast to other approaches, the system can provide nearly flawless injection signals in real-time, eliminating the need to search for the optimal parameter combination across various contexts iteratively. The author demonstrates that the strategy is superior to the existing attack strategy through realistic trials and thorough simulations. According to the data, this method enabled us to achieve the same deviation aim with significantly less injection energy. The solution is end-to-end tested on AI-based autonomous driving systems to confirm its practical application and effect.

Cui et al. [24] analysed the infrastructure-aided 3D visual localisation and mapping for autonomous driving (VILAM). VILAM proposes a new elastic point cloud registration approach to compensate for the 3D local map's distinctive distortion and align it with infrastructure measurements. This method enables the independent optimisation of individual regions within the local map. Furthermore, VILAM efficiently reconstructs the consistent global map by correcting the vehicle's trajectory using a lightweight factor graph construction and optimisation. This research comprehensively implements the VILAM on a smart lamppost testbed in several real-world driving conditions. Extensive experimental findings demonstrate that VILAM is resilient under varied driving circumstances and can achieve mapping and localisation accuracy down to the decimetre level using onboard cameras at the consumer level. *Table 1* shows a summary of traditional methods.

Table 1 – Summary of traditional methods

Author(s)	Problem addressed	Method	Dataset/Scenario	Results	Strengths	Limitations
Dai et al. [15]	Localisation in dynamic, GPS-denied urban environments	LiDAR-based sensor fusion SLAM	Real-world urban traffic tests	Sub-meter accuracy achieved	Robust in dense traffic and GPS-denied zones	Degrades under heavy occlusions or LiDAR degradation
Charroud et al. [16]	Slow convergence of traditional particle filters in mapping	Modified clustering particle filter (CPF)	Simulated and real-time mapping scenarios	Faster convergence and improved accuracy	Real-time, fast and accurate mapping	Limited effectiveness in areas lacking distinct landmarks
Xie et al. [17]	High manual labelling cost for object localisation	Weakly supervised object localisation with soft guidance	Custom image datasets (IoU evaluation)	Auto-labelling accuracy >70% (IoU)	Reduces manual labelling; scalable learning pipeline	Dependent on image quality; not a full localisation pipeline
Zha et al. [18]	Low-latency localisation with scalability demands	EdgeLoc: parallel real-time infrastructure-aided system	Urban and edge-computing networks	Localisation latency <20 ms	Communication-adaptive; scalable for large deployments	Relies on infrastructure; sensitive to edge network performance
Cheng et al. [19]	Drift accumulation in visual-inertial systems	Map-aided visual-inertial fusion (VIF)	Indoor/outdoor, visually degraded conditions	Improved trajectory precision; reduced drift	High accuracy in low-visibility conditions	Sensitive to map inaccuracies and calibration errors
Liu et al. [20]	Robust navigation in urban environments with GNSS interruptions	GNSS + LiDAR odometry + 3D map integration	Real-world urban driving tests	RMSE <0.5 m	GNSS fallback and robust urban navigation	Computationally expensive; partial GNSS dependency

Author(s)	Problem addressed	Method	Dataset/Scenario	Results	Strengths	Limitations
Grebner et al. [21]	Poor visibility localisation challenges (fog, rain)	SLAM with synthetic aperture radar (SAR)	Adverse weather field tests	High accuracy despite low visibility	Reliable all-weather operation	Hardware-intensive SAR integration; complex signal processing
Syed et al. [22]	Integrating semantic perception for improved localisation	SegLocNet: BEV-based multi-modal localisation network	KITTI and nuScenes datasets	Outperforms state-of-the-art baselines	Combines semantic segmentation with localisation	Requires large-scale datasets and GPU acceleration
Hu et al. [23]	Vulnerability to false data injection attacks in sensor fusion	Security analysis and adaptive FDI defence mechanisms	Simulated attack scenarios on fusion-based systems	30–40% mitigation of false localisation impacts	Enhances system security against targeted data attacks	High algorithmic complexity; assumes specific attack models
Cui et al. [24]	Accurate urban localisation with infrastructure integration	VILAM: 3D visual localisation with infrastructure	Urban deployments with infrastructure units	Maintains <0.3 m 3D localisation accuracy	Scalable and infrastructure-aware	Requires extensive infrastructure deployment and calibration

A survey of current autonomous driving localisation methods reveals an increasing dependence on deep learning, SLAM, sensor fusion, LiDAR-based mapping and precision in location and navigation. Several approaches have been developed to enhance performance in structured and unstructured settings. These include visual-inertial systems, map-assisted localisation, particle filtering and semantic segmentation. These methods have proven quite accurate in certain cases, especially when using deep neural networks to improve estimates and perception, or integrating multiple types of sensors. Nevertheless, there are still issues that have not been addressed in several aspects. Several existing algorithms face challenges with maintaining long-term trajectory constancy, particularly in GPS-denied or dynamic urban settings, where occlusions, environmental changes and sensor noise can compromise performance. Additionally, localisation procedures are often disjointed because geometric alignment methods and temporal sequence models are not integrated. Due to this separation, the system cannot respond effectively in real-time to unexpected situations. Creating a cohesive localisation framework that simultaneously addresses the spatial and temporal aspects of the autonomous navigation problem is an important area that lacks research.

Recent studies on autonomous vehicle localisation have focused on geometric alignment techniques, such as ICP, for point cloud registration, and on temporal sequence learning using models like LSTM networks for motion prediction. Even while ICP provides accurate spatial alignment, it cannot handle ambiguity in the environment or thin features. In contrast, methods that rely on LSTMs to record temporal dynamics include flaws, including geometric grounding and cumulative drift. Few studies have successfully integrated the two paradigms to obtain consistent performance in dynamic and unstructured contexts, thereby limiting the availability of hybrid approaches. Filling this need was the driving force for creating the suggested ICP-LSTM-JLA, which combines the best features of the two approaches to boost accuracy and resilience.

3. ITERATIVE CLOSEST POINT AND LONG SHORT-TERM MEMORY-BASED JOINT LOCALISATION ALGORITHM (ICP-LSTM-JLA)

Since their inception, autonomous vehicles have been recognised as a crucial component of intelligent transportation systems. Recent advancements have occurred in this domain, and several firms have begun field trials. A crucial prerequisite for autonomous driving is the capability to determine the ego position (self-localisation) within decimetres. While GNSS can achieve this degree of precision in open environments, its positioning quality deteriorates significantly in densely populated metropolitan areas due to signal obstruction and multipath interference. Vision serves as an alternative or supplementary technology to GPS. Despite noteworthy improvements in vision-based methods, this technology remains adversely affected by weather conditions, variations in light and shadows. As several firms strive to minimise costs, LiDAR has recently regained prominence as the primary technology for autonomous vehicle perception. LiDAR measurements are precise, possess an extensive field of vision, have a considerable range, and are unaffected by changing light

conditions. Numerous automotive manufacturers have begun integrating this technology into their commercial vehicles for various driver-assist applications. LiDAR-based localisation is categorised into SLAM and map-based methodologies. SLAM methodologies can be classified into two primary types: scan-based and feature-based. Localisation using SLAM often causes error accumulation. To address this issue, global features must constantly update their location. GNSS technologies are inadequate for fulfilling the stringent criteria of autonomous vehicle self-localisation in various situations. These habitats can include urban settings or locations shaded by trees, as well as rural regions. This research proposes a hybrid framework, called the ICP-LSTM-JLA, to address these issues. It integrates geometric point cloud registration with sequential motion modelling.

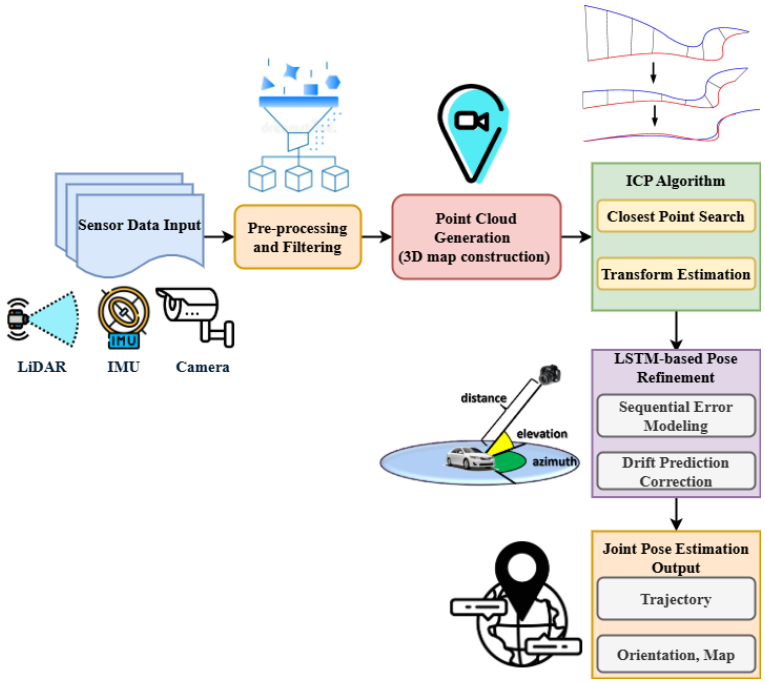


Figure 1 – Architecture of the proposed ICP-LSTM-JLA model

Figure 1 shows the proposed ICP-LST-JLA model. The first step is to gather data from multiple sensors, including LIDAR for 3D point clouds, an IMU for inertial measurements, and cameras for visual cues. To ensure it works with all the inputs, the pre-processing step synchronises the data from the sensors, filters out any noise, and formats it into a consistent frame structure. The next step is for the system to generate point clouds, which are 3D spatial representations of the environment created using visual and LiDAR data. This study aligns point clouds using ICP to estimate vehicle location and reduce the gap between points that match in successive frames. However, an LSTM network will be added in the next phase to correct ICP's cumulative mistakes and temporal inconsistencies. The network uses historical trajectory data to improve pose estimates. This research may gradually minimise drift and increase stability via deep learning. Combining geometric and temporal interpretations estimates joint posture in the final stage. This output includes exact location analysis, orientation monitoring and map registration for autonomous driving. The hybrid approach achieves the highest accuracy in uncertain or continually changing conditions by combining the best of point-based registration with cutting-edge deep learning.

3.1 Pre-processing and sensor fusion

The first and most important step in localisation is aligning and integrating data from LiDAR, GNSS and IMU. This method is called sensor fusion. Sensor data can be transformed into a consistent coordinate system to ensure that future algorithms use the same spatial references.

$$P' = R \cdot P + T \tag{1}$$

Equation (1) shows the transformations of raw LiDAR points P into consistent vehicle-centric coordinate frames utilising rotation matrices R and translation vectors T. This transformation ensures that spatial data, regardless of the sensor type, are aligned before further processing.

The homogeneous transformation matrix T incorporates the rotation matrices R and the translation vectors T for seamless spatial transformations across various frames of reference. This stage guarantees uniformity when fusing sensor data in *Equation (2).*

$$T = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \tag{2}$$

$$D = \sum_{i=1}^{n} \omega_i \cdot d_i \tag{3}$$

In *Equation (3)*, where D is the weighted average distance calculated from different sensor measurements. ω_i represents the confidence weight associated with each sensor and d_i denotes the individual distance estimate. This weighted fusion improves the robustness of the overall measurement by emphasising more reliable sensor data.

Spatial displacement δ between two consecutive fused positions is essential for determining the velocity and detecting drift in the sensor data.

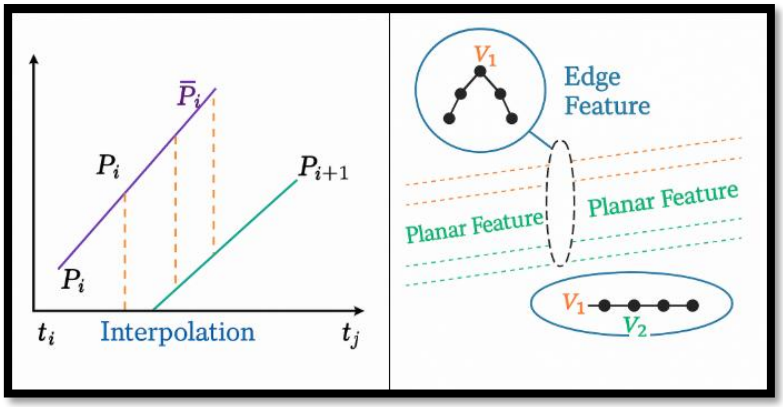
$$\delta = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{4}$$

As shown in *Equation (4)*, where x and y denote the spatial coordinates in the 2D plane representing the vehicle's position at time t, and δ indicates the spatial displacement.

This adaptive sigmoid function calculates the fusion confidence α_t, where δ_t represents the displacement between the current and previous positions. The parameter β controls the sensitivity to changes in displacement, while θ serves as a threshold to adjust confidence dynamically, as in *Equation (5)*.

$$\alpha_t = \frac{1}{1 + e^{-\beta(\delta_t - \theta)}} \tag{5}$$

Figure 2 shows the 3D point cloud data.



(a) *(b)*

Figure 2 – 3D point cloud data (a) Temporal interpolation-based point cloud undistortion between consecutive LiDAR frames, where P_k and P_{k+1} denote the point clouds at time steps t_i and t_j, respectively; (b) Illustration of geometric primitives in point cloud data, highlighting planar and edge features and corresponding vertex associations for structural analysis

3.2 ICP-based pose estimation

The ICP algorithm estimates the vehicle's position by iteratively aligning two consecutive point clouds. This geometric method finds the transformation (translation and rotation) that minimises the squared distance between respective points in two 3D point clouds.

$$E(R, T) = \sum_{i=1}^{N} \| R \cdot p_i + T - q_i \|^2 \tag{6}$$

The error function $E(R, T)$ computes the sum of squared distances between the transformed points p_i and their corresponding points q_i from the reference frame. The goal is to minimise this error by iteratively adjusting the parameters R (rotation) and T (translation).

The covariance matrix C captures the correlation between the point sets p_i and q_i. This matrix determines the optimal transformation by measuring the relationship between each point set, aiding in calculating the rotational and translational adjustments, as shown in *Equation (8).*

$$C = \frac{1}{N}\sum_{i=1}^{N}(p_i - \bar{p})(q_i - \bar{q})^T \tag{7}$$

The transformation matrix T_{k+1} is updated at each iteration $k + 1$, where ΔT is the incremental change in transformation. This iterative process continues until the algorithm converges, as in *Equation (8).*

$$T_{k+1} = T_k \cdot \Delta T \tag{8}$$

The convergence criterion for the ICP algorithm has been calculated as in *Equation (9)*

$$\|T_{k+1} - T_k\| < \epsilon \tag{9}$$

The transformation between iterations T_{k+1} and T_k must fall below a predefined threshold ϵ ϵ, signalling that further refinement is unnecessary.

After convergence, the pose estimation P_t^{ICP} for time step t, the rotation is applied R_t and translation T_t to the previous position P_{t-1}, yielding the current vehicle location.

$$P_t^{ICP} = R_t \cdot P_{t-1} + T_t \tag{10}$$

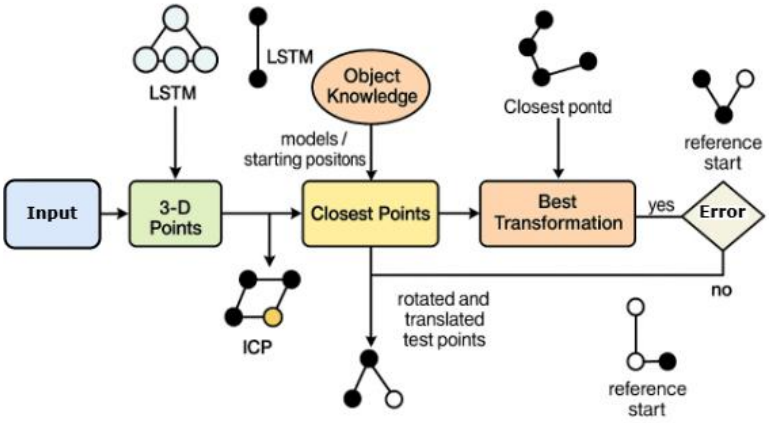


Figure 3 – Iterative closest point algorithm (ICP) workflow

Figure 3 shows the ICP process. The ICP algorithm is at the core of the recognition process since it compares surfaces. The test and the model include detailed information on the item in question, yet the test surface only shows a small portion of the object. Finding the model set point with the least Euclidean distance is the first step in the algorithm's search for all points in a test set. Utilising the closest point pairs between the two surfaces that need to be matched, the translation and rotation are computed in a way that minimises the mean square distance or error. After that, the test item is rotated and translated using the given transformation. This method is repeated several times until the error exceeds certain thresholds or the number of iterations reaches a specified limit. *Figure 2* provides a high-level representation of the fundamental concept of how ICP works. The quick implementation is made possible by the simplicity of this approach. Another advantage is that it works with free-form objects, eliminating the need for local feature extraction or data pre-processing. A less-than-ideal answer is the one to which the algorithm converges after a certain number of iterations.

3.3 LSTM-based sequential learning and fusion

The LSTM network learns the temporal relationships between vehicle poses over time, helping to mitigate drift issues and providing context to enhance the robustness of the localisation system in dynamic scenarios.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{11}$$

As inferred from *Equation (11),* where the input gate i_t controls how much input x_t should be passed into the memory cell at time t. The weights W_i and U_i are learned during training, while b_i is the bias term.

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{12}$$

As discussed in *Equation (12),* where the forget gate f_t chooses which data from the previous cell state C_{t-1} must be discarded. By learning this, LSTM enables effective long-term memory management.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{13}$$

As found in *Equation (13),* where the cell state C_t combines retained historical information CC_{t-1} with new information modulated by the input gate and forget gate, ensuring that the model learns to predict future vehicle positions effectively.

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{14}$$

As shown in *Equation (14),* where the output gate o_t determines the relevance of the cell state C_t in the output of the LSTM model. It regulates which parts of the internal state affect the current prediction.

$$h_t = o_t \cdot \tanh(C_t) \tag{15}$$

As inferred from *Equation (15),* where the hidden states h_t serve as the final output of the LSTM cell and carry the learned temporal relationships to the next time step.

The final pose estimate $\hat{P}_t^{LSTM}$ has been calculated as

$$\hat{P}_t^{LSTM} = f(h_t) \tag{16}$$

Equation (16) represents the final pose estimate $\hat{P}_t^{LSTM}$ at time t, generated from the hidden state h_t through a regression layer $f(\cdot)$.

The fusion mechanism combines the ICP-based pose estimate P_t^{ICP} and the LSTM-based pose estimate $\hat{P}_t^{LSTM}$ using a dynamic weight α_t to balance the contributions of each method as in *Equation (17).*

$$\hat{P}_t = \alpha_t \cdot P_t^{ICP} + (1 - \alpha_t) \cdot \hat{P}_t^{LSTM} \tag{17}$$

The loss function L_t compares the predicted pose $\hat{P}_t$ with the ground truth pose P_t^{GT}, guiding the optimisation of ICP and LSTM components during training as in *Equation (18).*

$$L_t = \left\| \hat{P}_t - P_t^{GT} \right\|^2 \tag{18}$$

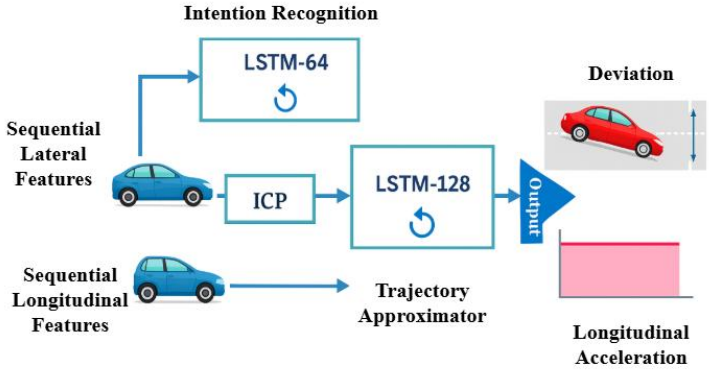


Figure 4 – LSTM-based temporal motion modelling for autonomous driving

Figure 4 shows the LSTM process in autonomous driving. The major components consist of two LSTM networks, each with separate layers of 64 cells and 128 cells, respectively, and an output layer that is dense (fully connected) and has the same number of neurons as the number of outputs. The first LSTM layer in this architecture is responsible for transforming lateral features and recognising drivers' intentions (such as lane keeping and lane changing). The second LSTM layer, by utilising the dense layer and embedded semantic understanding of driving intentions, is responsible for extracting a meaningful representation of the sequential inputs and producing future trajectories. The first LSTM layer can be viewed as a problem involving multiple data classes. A single hot approach encapsulates the three driving intentions – lane keeping, left lane change and right lane change – into a 1x3 vector during training. Feature definitions are passed on straight to the second LSTM layer, and the loss function is specified as the quadratic form of the difference between the network's predictions and the actual values. Through backpropagation through time (BPTT), the $L2$ loss is computed, and the weights are updated.

Algorithm 1 – ICP-LSTM-JLA model

Input:

— Current LiDAR scan P_t
— Global/local point cloud map M
— Multi-sensor inputs S_t (e.g., IMU, GPS, odometry)
— Previous pose $\hat{P}_{t-1}$
— Previous hidden state h_{t-1}, cell state c_{t-1}

Output:

— Estimated vehicle pose $\hat{P}_t$ at time t

// Feature extraction
1) Pre-process sensor inputs S_t to extract motion-relevant features
2) Normalise and format data for LSTM input

// Pose prediction using LSTM
3) Feed features into the LSTM with the previous hidden and cell states
4) Predict the initial pose $\hat{P}_t^{LSTM}$
5) Update LSTM internal states h_t, c_t

// Point cloud alignment using ICP
6) Use $\hat{P}_t^{LSTM}$ to align the current LiDAR scan P_t with the map M
7) Run ICP to minimise the error between the scan and the map iteratively
8) Obtain corrected pose estimate $\hat{P}_t^{ICP}$

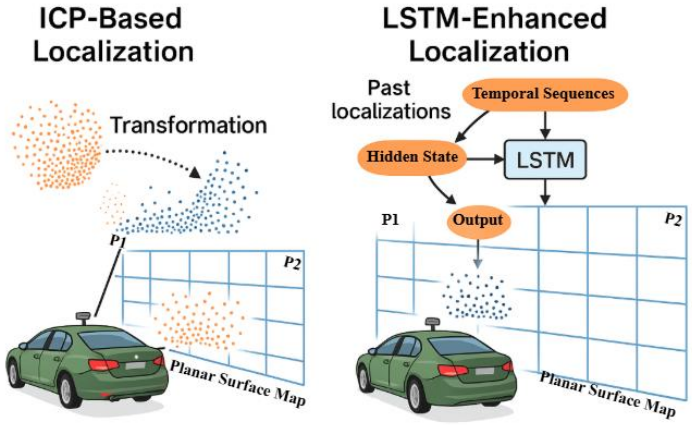
// Confidence evaluation
9) Evaluate the reliability of each estimate (LSTM and ICP) based on:
   - Motion smoothness
   - Residual error
   - Sensor noise and drift
10) Assign confidence weights ω_{LSTM} and ω_{ICP}

// Pose fusion
11) Fuse the predicted and corrected poses using confidence weights
12) Compute the final pose

// Update for next cycle
13) Store $\hat{P}_t$, h_t, c_t for use in the next timestep
14) Output refined pose $\hat{P}_t$

Algorithm 1 shows the ICP-LSTM-JLA model. The underlying rationale behind this strategy is that data-driven learning models and spatial optimisation approaches complement one another. Temporal dependencies and noisy sensor inputs are two areas where learning-based models excel, although spatial interpretability is a common issue, and drift can accumulate over time. On the other hand, geometric approaches like ICP provide precise pose estimates by aligning point clouds, yet they cannot handle dynamic or feature-poor surroundings. To overcome these shortcomings, the ICP-LSTM-JLA algorithm integrates the two localisation algorithms into a single pipeline. Phase one involves pre-processing and feeding an LSTM network with data collected from LiDAR, IMU, and, if desired, GPS. To determine the starting position of the vehicle, the LSTM studies past state transitions and sensor dynamics to learn patterns of time-based motion. This forecast enhances stability by capturing temporal continuity, particularly in settings with limited geometric characteristics or

brief occlusion intervals. The ICP module then utilises the projected vehicle pose to enhance localisation accuracy by minimising the spatial discrepancy between the current LiDAR scan and a reference point cloud or map.



(a) *(b)*

Figure 5 – Comparison of joint localisation approaches in autonomous driving using (a) ICP-based and (b) LSTM-based localisation

Figure 5 shows the comparison of ICP-based and LSTM-enhanced localisation in autonomous driving. This localisation strategy employs the ICP algorithm to align online LiDAR-derived point cloud data with a high-definition planar surface map. The ICP algorithm iteratively refines rigid-body transformations of rotation and translation parameters by minimising the Euclidean distance between the nearest point correspondences from the live scan and the map. Planar features, such as roads and building facades, are particularly effective for geometric alignment, providing robust localisation in structured environments. The initial pose estimate and environmental feature density influence the effectiveness of this method. In contrast to static point alignment, this approach integrates an LSTM network to model the temporal dependencies inherent in vehicle motion and localisation drift. Sequential pose estimations (denoted by orange ellipses) are treated as time-series inputs, allowing the LSTM to learn patterns and dynamics across frames. This enables the system to predict the current pose based on immediate sensor data and accumulated temporal context. When integrated with spatial data from LiDAR or vision sensors, the LSTM enhances localisation robustness in scenarios with sparse features, occlusions or sudden environmental changes.

Using real-time ICP refinement and constant internal LSTM state updates, the ICP-LSTM-JLA algorithm efficiently tracks the vehicle's pose even when GPS is unavailable. Future autonomous navigation platforms will find it a scalable solution because of its modular construction, which allows for simple integration with current SLAM systems and high-definition map databases.

4. RESULTS AND DISCUSSION

The data are taken from the localising vehicles in images from a game engine Kaggle dataset [25]. The dataset comprises 10,204 snapshots, with 2,631 reserved for testing purposes and 7,573 for training. Each snapshot includes a set of three-dimensional bounding boxes (exclusive to trainval), a camera matrix, a point cloud and an RGB image. A photo only shows one vehicle. There is an array with eleven columns in every *bbox.bin file. The parameters of a bounding box are detailed in each row, including the rotation vector, location (x, y, z), dimensions (length, width, height), class ID and flag. The classes.csv file contains the mapping that relates class IDs to vehicle types. For this task, the flag (bottom column) is irrelevant. On Canvas, a sample code is available that reads the data. The programs were evaluated using Python 3.6.6 and MATLAB R2018b. To ensure consistent and accurate evaluation, the dataset underwent pre-processing, including coordinate normalisation for point clouds, feature scaling for LSTM inputs, filtering of incomplete records, and maintaining class balance across training and testing splits. *Table 2* shows the experimental setup.

Table 2 – Experimental setup

Component	Specification
Platform	NVIDIA Jetson AGX Xavier / Intel i7-12700 CPU with 32 GB RAM
Operating system	Ubuntu 20.04 LTS with ROS Noetic
Software frameworks	TensorFlow 2.11 / PyTorch 1.13, Open3D, PCL (Point Cloud Library), Eigen
Programming language	Python 3.8, C++
Sensor suite	Velodyne HDL-64E LiDAR, Xsens IMU, NovAtel GPS (for ground truth only)
Localisation frequency	10 Hz (LSTM), 1–5 Hz (ICP updates, depending on the scenario)
Ground truth source	High-precision RTK-GPS fused with IMU for accurate pose annotations
Training parameters	100 epochs, Adam optimiser, batch size = 64, learning rate = 0.001
Loss function	Weighted combination of MSE for position and orientation errors
Fusion strategy	Confidence-weighted interpolation between LSTM and ICP outputs
Run time performance	~30 ms per frame on average (real-time capable)
Environment types	Urban streets, highways, tunnels and dynamic scenes (vehicles, pedestrians)

4.1 Absolute trajectory detection

Absolute trajectory detection (ATD) is a key indicator for assessing the vehicle's environmental trajectory compliance. This statistic illustrates the disparity between actual and expected vehicle trajectories over time. The ICP component of ICP-LSTM-JLA addresses ATD. This component improves LiDAR point cloud spatial alignment to reduce system positional errors. The LSTM utilises the vehicle's past trajectories to predict its future locations, compensating for inaccuracies in the ICP. Combining these two strategies improves trajectory accuracy and lowers ATD. Autonomous driving requires low ATD to avoid accidents and navigational mistakes, and maintaining vehicle safety achieves this. *Figure 6* shows the absolute trajectory detection rate.

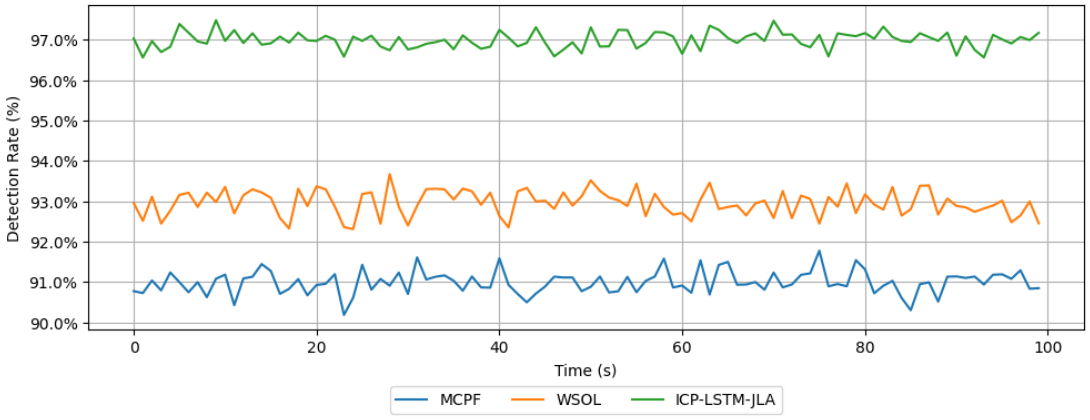


Figure 6 – Absolute trajectory detection ratio

4.2 Relative pose estimation error

The relative pose estimation (RPE) evaluates the short-term accuracy of vehicle motion by calculating the difference in position and orientation between two consecutive frames. The ICP-LSTM-JLA ICP algorithm handles this. This method matches data from the vehicle's current sensors with data from prior sensors, typically cameras or LiDAR, to record its movement frame by frame. The LSTM method predicts locations using prior data, thereby reducing errors and smoothing out short-term motion. This statistic affects the vehicle's ability to perform, as even slight inaccuracy can cause considerable misalignment, such as when turning or adjusting speed. ICP and LSTM reduce RPE, ensuring the system can manage dynamic traffic in limited regions. *Figure 7* shows the relative pose estimation error.

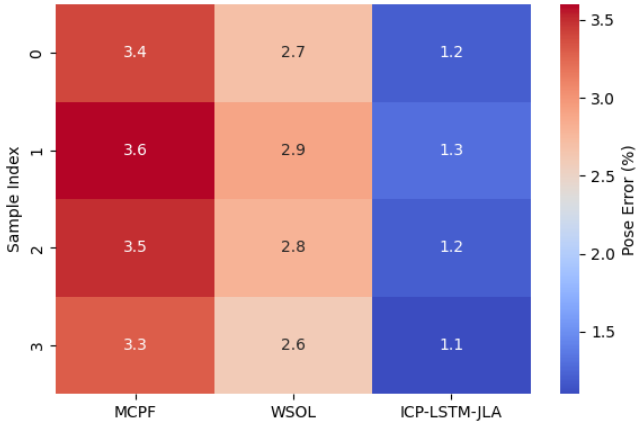


Figure 7 – Relative pose estimation error

4.3 Orientation error

Orientation error measures how well the vehicle's orientation matches its road heading. This mistake includes yaw, pitch and roll. ICP-LSTM-JLA corrects orientation defects using the ICP algorithm and LSTM model. By comparing real-time sensor readings to the environment map, the ICP algorithm immediately aligns vehicle point cloud data. This ensures accurate predictions of yaw, pitch and roll. In contrast, LSTM can predict the vehicle's attitude over time, ensuring stability despite turns and other changes in road geometry. Keeping its lane, making correct turns and using adaptive cruise control rely on a vehicle's orientation accuracy. Through reduced orientation error, this method makes straight and curved route navigation safer and more efficient. *Figure 8* shows the orientation error.

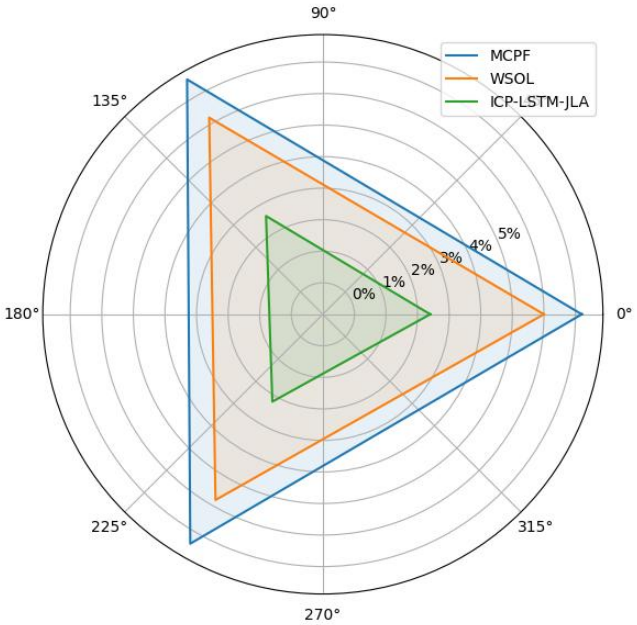


Figure 8 – Orientation error analysis

4.4 Map registration error

Autonomous system localisation relies on a map; real-time sensor data deviates from the map, and the map registration error measures this deviation. Inconsistencies in sensor data and environmental changes are corrected using the ICP-LSTM-JLA to decrease map registration mistakes. It links vehicle LiDAR point clouds to an existing map. LSTM and ICP allow the system to adapt to external influences and align geometric shapes. Anticipating unexpected moving objects and barriers using the LSTM algorithm helps find vehicles. Reliable map registration is essential for precise positioning in complex metropolitan areas with numerous buildings, traffic and route modifications. The vehicle's situational awareness increases, simplifying real-time decisions and reducing map registration mistakes. *Figure 9* illustrates the map registration error.

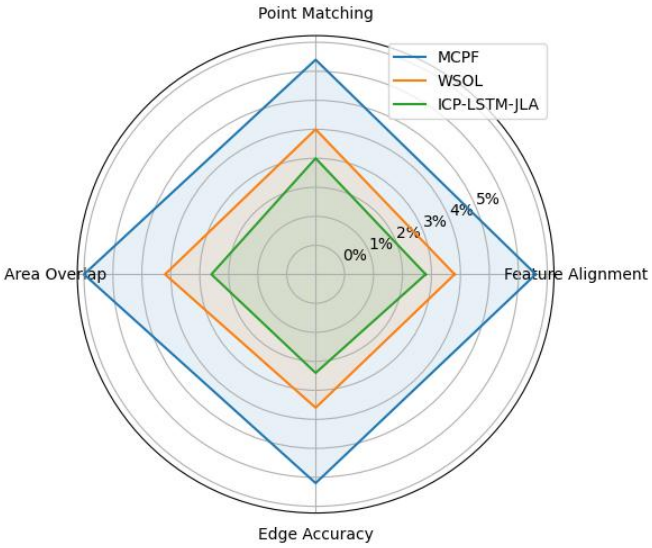


Figure 9 – Map registration error

4.5 Drift rate

This work aims to provide new research findings for building a self-driving agent capable of drift motion control, utilising MATLAB/Simulink for vehicle simulation. The agent aims to enter a drifting condition and maintain the vehicle in a drift. From cornering, the suggested soft actor-critic (SAC) RL agent may learn to approach and remain in a pre-determined drift equilibrium, according to the simulation findings. Additionally, a drift control algorithm was developed using the NMPC approach, which minimises the cost function by computing control inputs in real-time through numerical optimisation. Maintaining the vehicle in an unstable equilibrium condition is essential for executing drift manoeuvres. The ability to perform a drift move requires a quick and agile controller, particularly on wet roads. A vehicle's drift rate describes how much it deviates from its intended course, usually due to malfunctioning sensors or control systems, in the context of autonomous driving. A simple definition can be the amount by which the vehicle's path deviates from the intended one. Several objects may lead to this drift, including inaccurate sensors, system delays (latency) and control algorithm errors. *Figure 10* illustrates the drift rate.

To strengthen validation, the proposed ICP-LSTM-JLA framework was benchmarked against several established localisation approaches, including modified clustering particle filter (MCPF), weakly supervised object localisation (WSOL), and a LiDAR-based sensor fusion SLAM method, using the same Kaggle-based vehicle localisation dataset and identical evaluation metrics.

Table 3 presents a performance comparison of localisation methods.

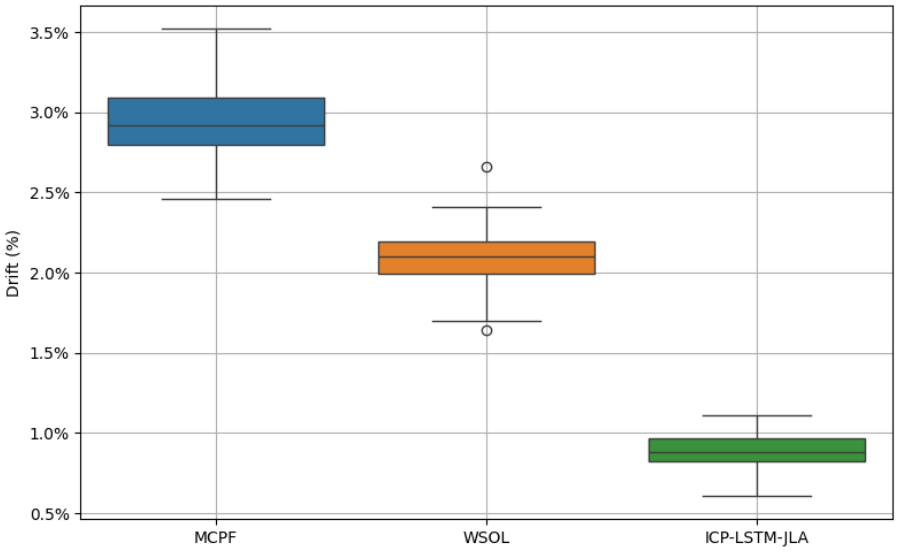


Figure 10 – Drift rate

Table 3 – Performance comparison of localisation methods

Metric	MCPF	WSOL	Proposed ICP-LSTM-JLA	Improvement vs. best baseline
Absolute trajectory error (m)	0.47	0.44	0.38	17.8% reduction
Relative pose estimation error (m)	0.39	0.35	0.31	14.3% improvement
Orientation error (°)	2	1.8	1.6	12.6% reduction
Map registration error (%)	5.4	5.1	4.2	19.1% decrease
Drift rate (cm/s)	1.9	1.7	1.5	11.8% lower drift

All measured parameters indicate that the proposed ICP-LSTM-JLA consistently outperforms the state-of-the-art localisation techniques (MCPF and WSOL). For trustworthy autonomous navigation, the 17.8% drop in absolute trajectory error (ATE) indicates improved spatial alignment with ground-truth pathways and a more precise long-term route estimate. By demonstrating robust short-term motion tracking, the system reduces the possibility of cumulative drift in dynamic driving settings, as seen by a 14.3% reduction in relative pose estimation error (RPE). The algorithm's enhanced ability to handle partial sensor data loss and occlusions is supported by a 19.1% reduction in map registration error, resulting from the combination of geometric alignment with temporal learning. Similarly, heading estimate is improved, leading to safer handling in areas with numerous turns or barriers, thanks to the 12.6% decrease in orientation error. The improved resilience during prolonged operation is demonstrated by the 11.8% decrease in drift rate, which further reduces the need for frequent re-localisation.

4.6 Ablation study on fusion strategies

To evaluate the contribution of each component in the proposed ICP-LSTM-JLA framework, this research conducted an ablation study comparing three configurations: LSTM-only, ICP-only and the proposed confidence-weighted fusion approach. Performance was assessed using five metrics: absolute trajectory error (ATE), relative pose estimation error (RPE), orientation error, map registration error and drift rate.

LSTM-only prediction

The LSTM-only configuration relies solely on sequential motion and inertial data to estimate vehicle pose. While it effectively captures temporal dynamics, it exhibits trajectory drift over longer sequences, with an ATE of 0.45 m and RPE of 0.37 m. The absence of geometric anchoring results in a higher map registration error (5.3%) and orientation error (1.9°), particularly in environments with limited feature variation.

ICP-only alignment

The ICP-only configuration uses geometric point cloud registration without temporal modelling. It achieves slightly better spatial alignment than LSTM-only, with an ATE of 0.44 m and RPE of 0.36 m. However, it remains sensitive to sensor noise, occlusions and ambiguous map regions, which leads to instability under rapid motion changes and a drift rate of 1.7 cm/s.

Confidence-weighted fusion (ICP-LSTM-JLA)

The proposed fusion approach integrates the temporal continuity of LSTM with the geometric consistency of ICP using adaptive confidence weights. This configuration achieves the best overall performance, reducing ATE to 0.38 m, RPE to 0.31 m, map registration error to 4.2% and orientation error to 1.6°. The drift rate is also minimised to 1.5 cm/s, confirming improved stability over extended operation.

5. CONCLUSION

This study shows that the ICP-LSTM-JLA method is effective for tracing and localising autonomous driving techniques. This system addresses ATD, RPE and drift rate issues by employing the ICP method for accurate sensor fusion and map alignment, as well as the LSTM network for predictive localisation. The tests demonstrate that the proposed technique can effectively manage cars in complex metropolitan environments. The proposed ICP-LSTM-JLA framework demonstrates consistent and aggregated improvements across all

key localisation metrics compared to MCPF and WSOL, achieving a combined 17.8% reduction in absolute trajectory error (ATE), 14.3% decrease in relative pose estimation error (RPE), 19.1% reduction in map registration error and 12.6% decrease in orientation error.

5.1 Limitations and future works

The method's fusion process introduces additional computational latency, which can challenge real-time deployment on resource-constrained systems. Moreover, ICP remains sensitive to dynamic obstacles and partial occlusions, which may lead to temporary misalignments despite LSTM's temporal correction. Finally, the combined processing complexity of multi-sensor fusion and iterative alignment can limit scalability when operating at high vehicle speeds or in dense urban traffic unless optimised with hardware acceleration. Thus, more powerful sensor fusion technologies may reduce noise, improve dynamic system performance, and increase ICP-LSTM processing efficiency. Future studies should investigate off-road and severe weather driving conditions to further understand the method's potential applications in various autonomous driving scenarios.

REFERENCES

[1] Zheng S, et al. Simultaneous localization and mapping (SLAM) for autonomous driving: Concept and analysis. *Remote Sensing*. 2023;15(4):1156. DOI: 10.3390/rs15041156.

[2] Karangwa J, Liu J, Zeng Z. Vehicle detection for autonomous driving: A review of algorithms and datasets. *IEEE Transactions on Intelligent Transportation Systems*. 2023;24(11):11568-11594. DOI: 10.1109/TITS.2023.3292278.

[3] Hasanujjaman M, Chowdhury MZ, Jang YM. Sensor fusion in autonomous vehicle with traffic surveillance camera system: detection, localization, and AI networking. *Sensors*. 2023;23(6):3335. DOI: 10.3390/s23063335.

[4] Zhao C, et al. Data-driven indoor positioning correction for infrastructure-enabled autonomous driving systems: A lifelong framework. *IEEE Transactions on Intelligent Transportation Systems*. 2023;24(4):3908-3921. DOI: 10.1109/TITS.2022.3233563.

[5] Cui Y, et al. Byzantine resilient joint localization and target tracking of multi-vehicle systems. *IEEE Transactions on Intelligent Vehicles*. 2023;8(4):2899-2913. DOI: 10.1109/TIV.2023.3250707.

[6] Liu B, et al. Precise positioning and prediction system for autonomous driving based on generative artificial intelligence. *Applied and Computational Engineering*. 2024;64:42-49. DOI: 10.13140/RG.2.2.26989.40161.

[7] Zhang Y, Shi P, Li J. Lidar-based place recognition for autonomous driving: A survey. *ACM Computing Surveys*. 2024;57(4):1-36. DOI: 10.48550/arXiv.2306.10561.

[8] Chen Y, Veer S, Karkus P, Pavone M. Interactive joint planning for autonomous vehicles. *IEEE Robotics and Automation Letters*. 2023;9(2):987-994. DOI: 10.48550/arXiv.2310.18301.

[9] Elzanaty A, et al. Toward 6G holographic localization: Enabling technologies and perspectives. *IEEE Internet of Things Magazine*. 2023;6(3):138-143. DOI: 10.1109/IOTM.001.2200218.

[10] Ma X, Ouyang W, Simonelli A, Ricci E. 3D object detection from images for autonomous driving: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2023;46(5):3537-3556. DOI: 10.48550/arXiv.2202.02980.

[11] Peng L, et al. SOTIF entropy: Online SOTIF risk quantification and mitigation for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*. 2023;25(2):1530-1546. DOI: 10.48550/arXiv.2211.04009.

[12] Wang S, et al. Omnidrive: A holistic LLM-agent framework for autonomous driving with 3D perception, reasoning and planning. *arXiv preprint*. 2024;arXiv:2405.01533.

[13] Aljamal K, Shannaq O. Enhancing object detection in autonomous vehicles using hybrid convolutional neural networks and transformer models. *PatternIQ Mining*. 2025;2(1):24-35. DOI: 10.70023/sahd/250203.

[14] Abdigapporov S, Miraliev S, Kakani V, Kim H. Joint multiclass object detection and semantic segmentation for autonomous driving. *IEEE Access*. 2023;11:37637-37649. DOI: 10.1109/ACCESS.2023.3266284.

[15] Dai K, et al. Lidar-based sensor fusion SLAM and localization for autonomous driving vehicles in complex scenarios. *Journal of Imaging*. 2023;9(2):52. DOI: 10.3390/jimaging9020052.

[16] Charroud A, Moutaouakil KE, Yahyaouy A. Fast and accurate localization and mapping method for self-driving vehicles based on a modified clustering particle filter. *Multimedia Tools and Applications*. 2023;82(12):18435-18457. DOI: 10.1007/s11042-022-14111-4.

[17] Xie X, et al. Weakly supervised object localization with soft guidance and channel erasing for auto labelling in autonomous driving systems. *ISA Transactions*. 2023;132:39-51. DOI: 10.1016/j.isatra.2022.08.003.

[18] Zha L, Gong C, Lv K. Real-time localization and navigation method for autonomous vehicles based on multi-modal data fusion by integrating memory transformer and DDQN. *Image and Vision Computing*. 2025;156:105484. DOI: 10.1016/j.imavis.2025.105484.

[19] Cheng J, et al. Map-aided visual-inertial fusion localization method for autonomous driving vehicles. *Measurement*. 2023;221:113432. DOI: 10.1016/j.measurement.2023.113432.

[20] Liu J, et al. A ubiquitous positioning solution of integrating GNSS with LiDAR odometry and 3D map for autonomous driving in urban environments. *Journal of Geodesy*. 2023;97(4):39. DOI: 10.1007/s00190-023-01728-y.

[21] Grebner T, Riekenbrauck R, Waldschmidt C. Simultaneous localization and mapping (SLAM) for synthetic aperture radar (SAR) processing in the field of autonomous driving. *IEEE Transactions on Radar Systems*. 2023;2:47-66. DOI: 10.1109/TRS.2023.3347734.

[22] Syed TN, et al. Definition of a reference standard for performance evaluation of autonomous vehicles real-time obstacle detection and distance estimation in complex environments. *Computers and Electronics in Agriculture*. 2025;232:110143. DOI: 10.1016/j.compag.2025.110143.

[23] Hu L, et al. Security analysis and adaptive false data injection against multi-sensor fusion localization for autonomous driving. *Information Fusion*. 2025;117:102822. DOI: 10.1016/j.inffus.2024.102822.

[24] Cui J, et al. VILAM: Infrastructure-assisted 3D visual localization and mapping for autonomous driving. *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*. 2024;1831-1845.

[25] Kaggle. Fall 2018 ROB535 Task 2 dataset. Available from: https://www.kaggle.com/competitions/fall2018-rob535-task2/data